基于MindX SDK开发目标检测应用



华为技术有限公司

目录

[1 实验环境介绍 5](#_Toc117841757)

[1.1 实验介绍 5](#_Toc117841758)

[1.1.1 关于本实验 5](#_Toc117841759)

[1.1.2 实验环境 5](#_Toc117841760)

[1.1.3 实验目的 6](#_Toc117841761)

[1.1.4 实验清单 6](#_Toc117841762)

[2 基于MindX SDK开发目标检测应用 7](#_Toc117841763)

[2.1 实验流程 7](#_Toc117841764)

[2.2 环境准备 7](#_Toc117841765)

[2.3 配置工程 7](#_Toc117841766)

[2.3.1 启动MindStudio 7](#_Toc117841767)

[2.3.2 安装MindX SDK 10](#_Toc117841768)

[2.4 预置离线模型和测试图片 13](#_Toc117841769)

[2.4.1 预制离线模型 13](#_Toc117841770)

[2.4.2 预置测试图片 14](#_Toc117841771)

[2.5 业务流程编排 15](#_Toc117841772)

[2.6 配置插件属性 16](#_Toc117841773)

[2.7 编写应用代码 18](#_Toc117841774)

[2.8 编译运行 19](#_Toc117841775)

[2.8.1 编译 19](#_Toc117841776)

[2.8.2 运行 20](#_Toc117841777)

[2.9 结果可视化 21](#_Toc117841778)

[2.9.1 修改代码 21](#_Toc117841779)

[2.9.2 编译 24](#_Toc117841780)

[2.9.3 运行 25](#_Toc117841781)

[2.10 实验小结 26](#_Toc117841782)

[3 附录：环境准备 27](#_Toc117841783)

[3.1 获取镜像 27](#_Toc117841784)

[3.2 创建ECS 弹性服务器 28](#_Toc117841785)

[3.2.1 创建ECS弹性云服务器 28](#_Toc117841786)

[3.2.2 MobaXterm连接ECS 33](#_Toc117841787)

[3.3 关闭ECS弹性云服务器 35](#_Toc117841788)

|  |
| --- |
| 版权所有 © 华为技术有限公司 2021。 保留一切权利。  非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。  商标声明  C:\Users\jwx341670\Desktop\华为标志 Huawei Logo 2018\竖版标志Vertical Version\PNG\HW_POS_RBG_Vertical-150ppi.png 和其他华为商标均为华为技术有限公司的商标。  本文档提及的其他所有商标或注册商标，由各自的所有人拥有。  注意  您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。  由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。 |

|  |  |
| --- | --- |
| 华为技术有限公司 | |
| 地址： | 深圳市龙岗区坂田华为总部办公楼 邮编：518129 |
| 网址： | http://[e](http://e.huawei.com/).huawei.com |

# 实验环境介绍

## 实验介绍

### 关于本实验

通过业务流编排、配置插件属性、编写应用代码、预置离线模型与测试图片、结果可视化全流程展示如何使用MindX SDK开发目标检测应用。

### 实验环境

弹性云服务器（ECS）、MobaXterm、MindStudio、mxvision或mxmanufacture。

弹性云服务器（Elastic Cloud Server,ECS） 是由CPU、内存、操作系统、云硬盘组成的基础的计算组件。弹性云服务器创建成功后，您可以像使用自己的本地PC或服务器一样，在云上使用弹性云服务器。

MobaXterm (<https://mobaxterm.mobatek.net/>) 远程链接工具。

MindStudio 是一套基于IntelliJ框架的开发工具链平台，提供了应用开发、调试、模型转换功能，同时还提供了网络移植、优化和分析功能，为用户开发应用程序带来了极大的便利。

MindX SDK致力于简化昇腾芯片推理业务开发过程，降低使用昇腾芯片开发的门槛。SDK采用模块化的设计理念，将业务流程中的各个功能单元封装成独立的插件，通过插件的串接快速构建业务。开发者可选用mxvision完成本实验。

昇腾软件栈

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **软件** | **版本** | **说明** | **获取方式** | **运行环境** |
| mxvision | 2.0.1 | mxvision软件包 |  | ECS |
| Ascend-CANN-toolkit | 20.2.0 | Ascend-cann-toolkit开发套件包 |  |
| Ascend Driver | 20.2.0 | Ascend310芯片驱动 |  |
| 操作系统 | Ubuntu 18.04.1 LTS | 操作系统 | Ubuntu官网获取 |

### 实验目的

通过业务流编排、配置插件属性、编写应用代码、模型和图片、结果可视化全流程展示如何使用MindX SDK开发目标检测应用。

### 实验清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **实验** | **简述** | **难度** | **开发环境** | **运行环境** |
| 基于MindX SDK开发目标检测应用 | 基于MindX SDK开发目标检测应用 | 中级 | ECS | ECS |

# 基于MindX SDK开发目标检测应用

## 实验流程

1.环境准备

2.配置工程

3.预置离线模型与测试图片

4.业务流程编排

5.配置插件属性

6.编写应用代码

7.编译运行

8.结果可视化

## 环境准备

本实验需要获取镜像，购买ECS资源，并使用远程连接工具MobaXterm，详情见附录。

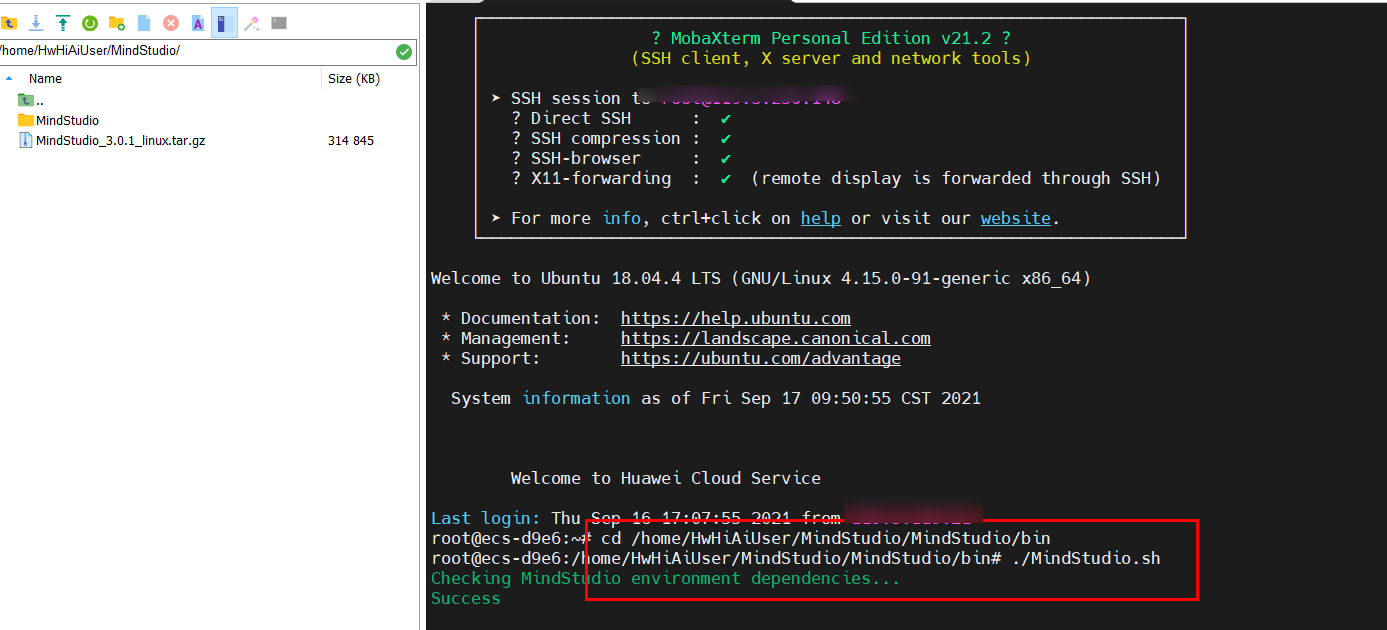
## 配置工程

### 启动MindStudio

输入以下命令启动：

cd /home/HwHiAiUser/MindStudio/MindStudio/bin

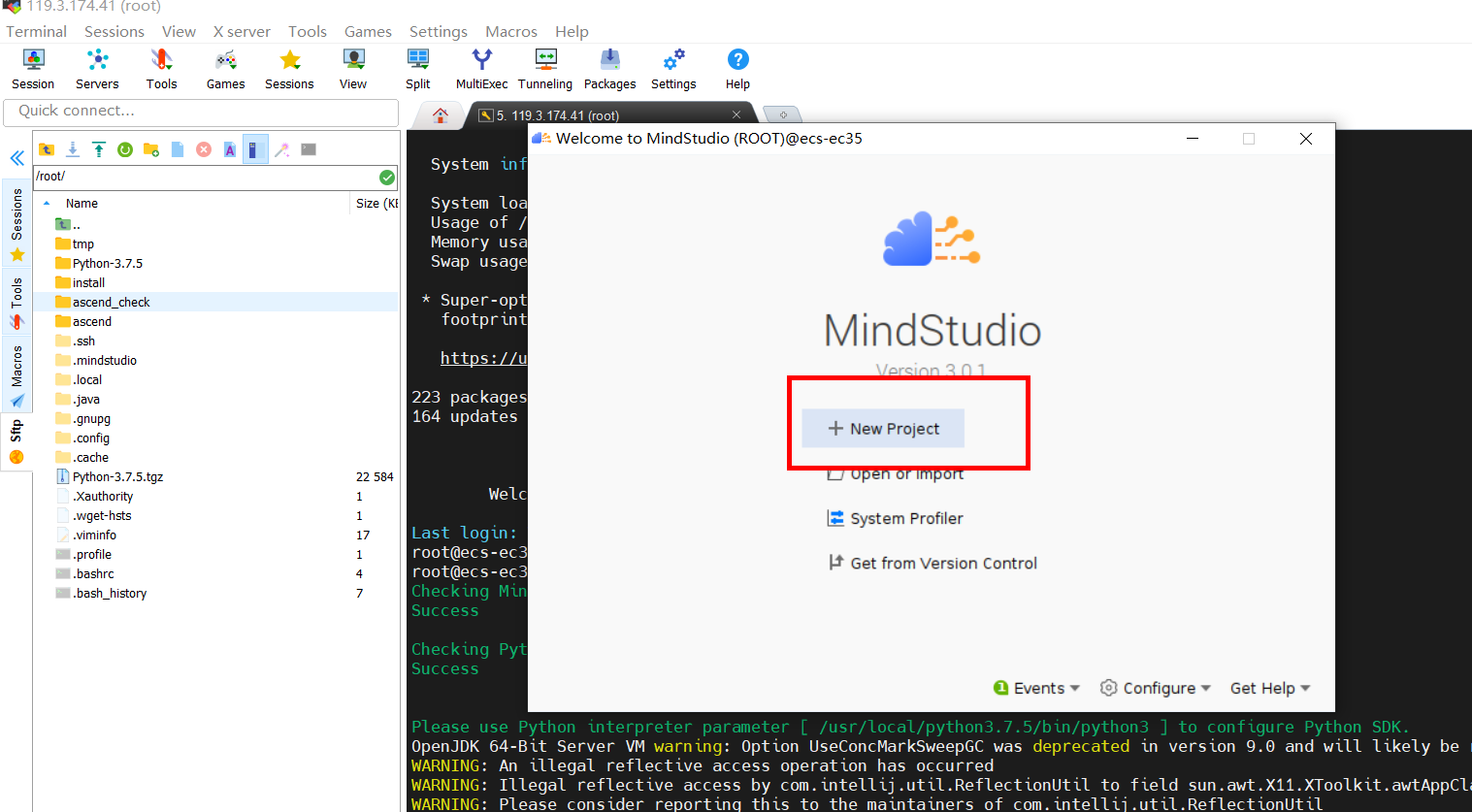
./MindStudio.sh



运行MindStudio

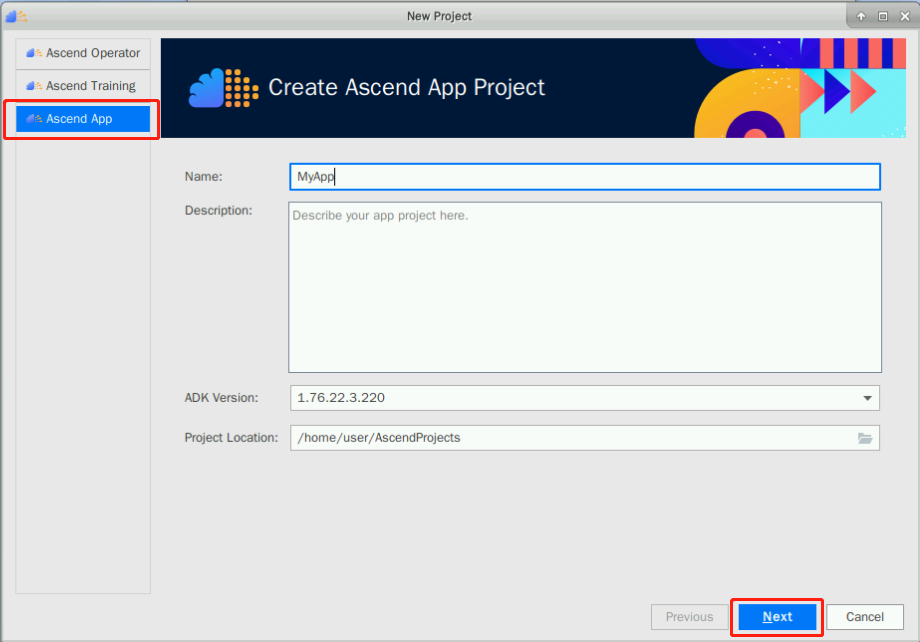
新建项目

启动成功后，选择“New Project”，创建一个新的项目



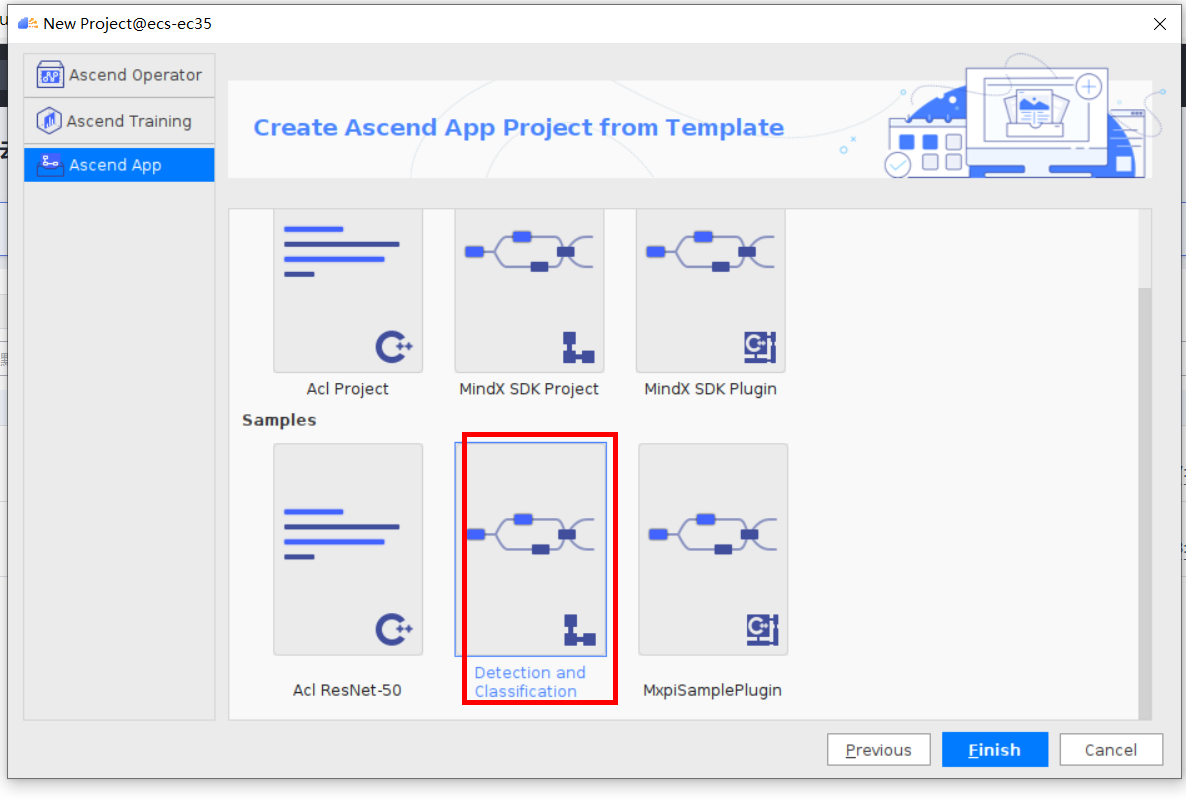
MindStudio 启动界面

选择“Ascend App”



新建project 界面

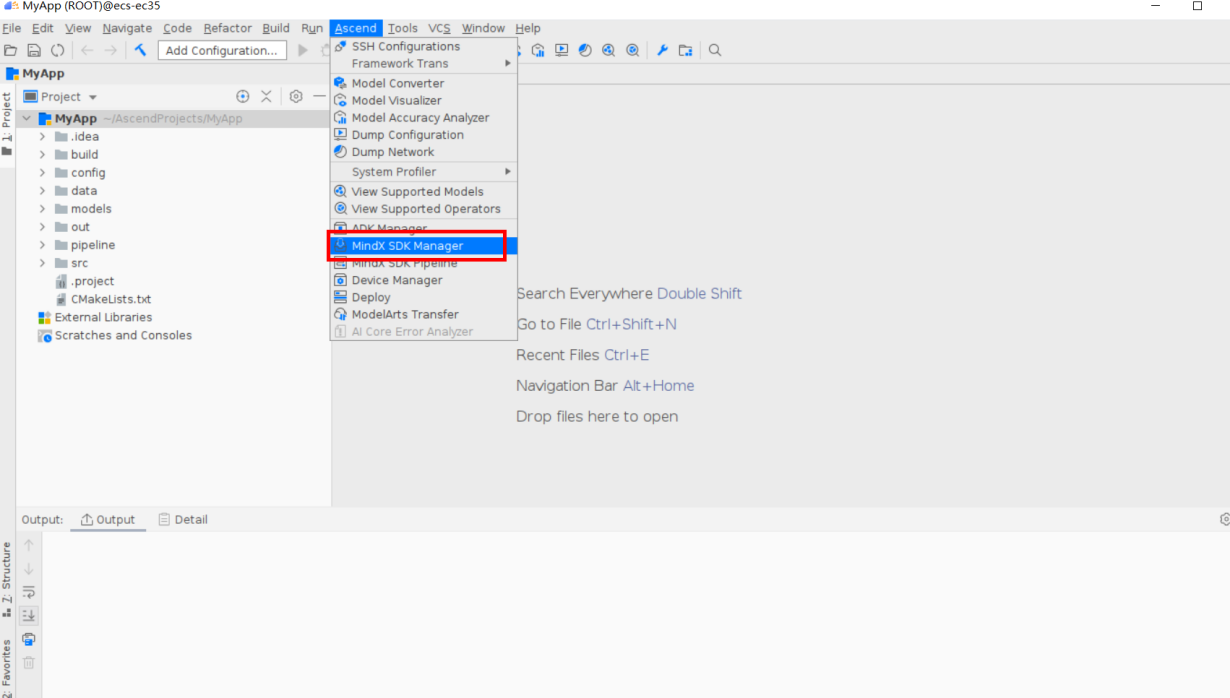
选择“Detection and Classification”



Ascend APP 创建界面

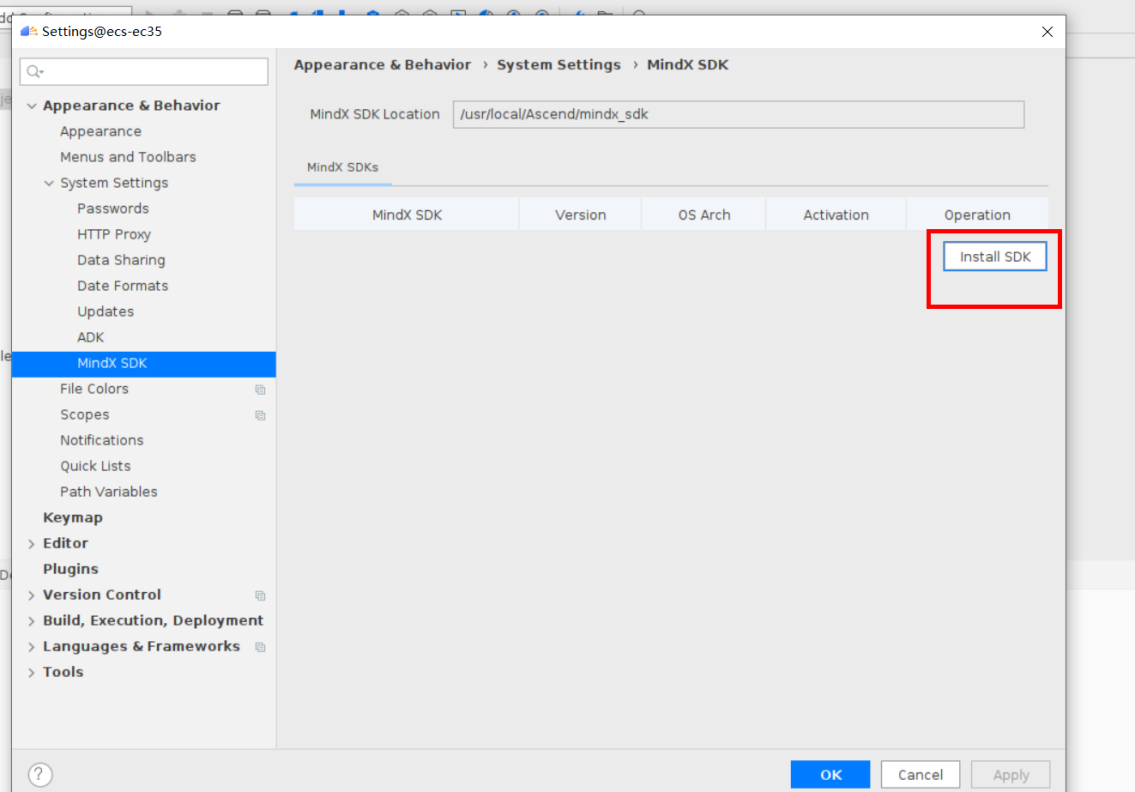
### 安装MindX SDK

在MindStudio的菜单栏依次点击“Ascend”->“MindX SDK Manager”如下



Mindx SDK Manager 界面

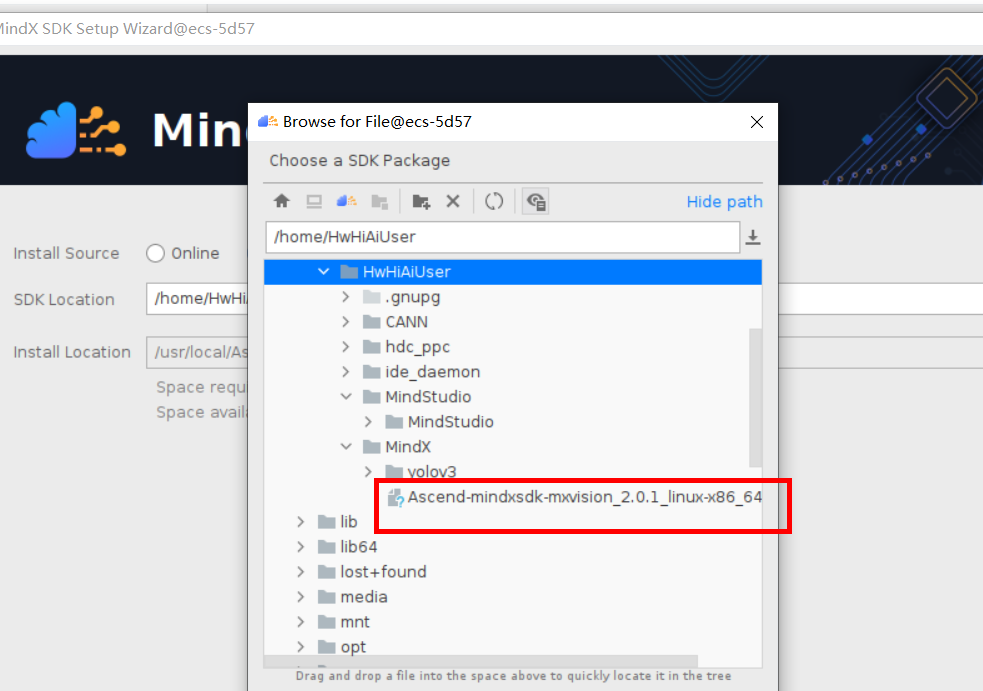
在弹出的窗口右侧选择“install SDK”



SDK 安装界面

选择“Disk”

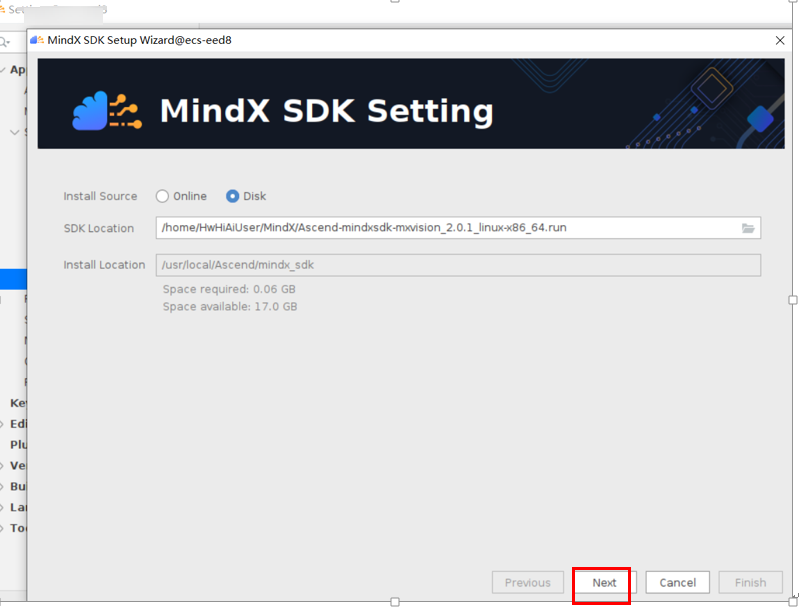
选择/home/HwHiAiUser/MindX/Ascend-mindxsdk-mxvision\_2.0.1\_linux-x86\_64.run包存放路径，如下图所示。



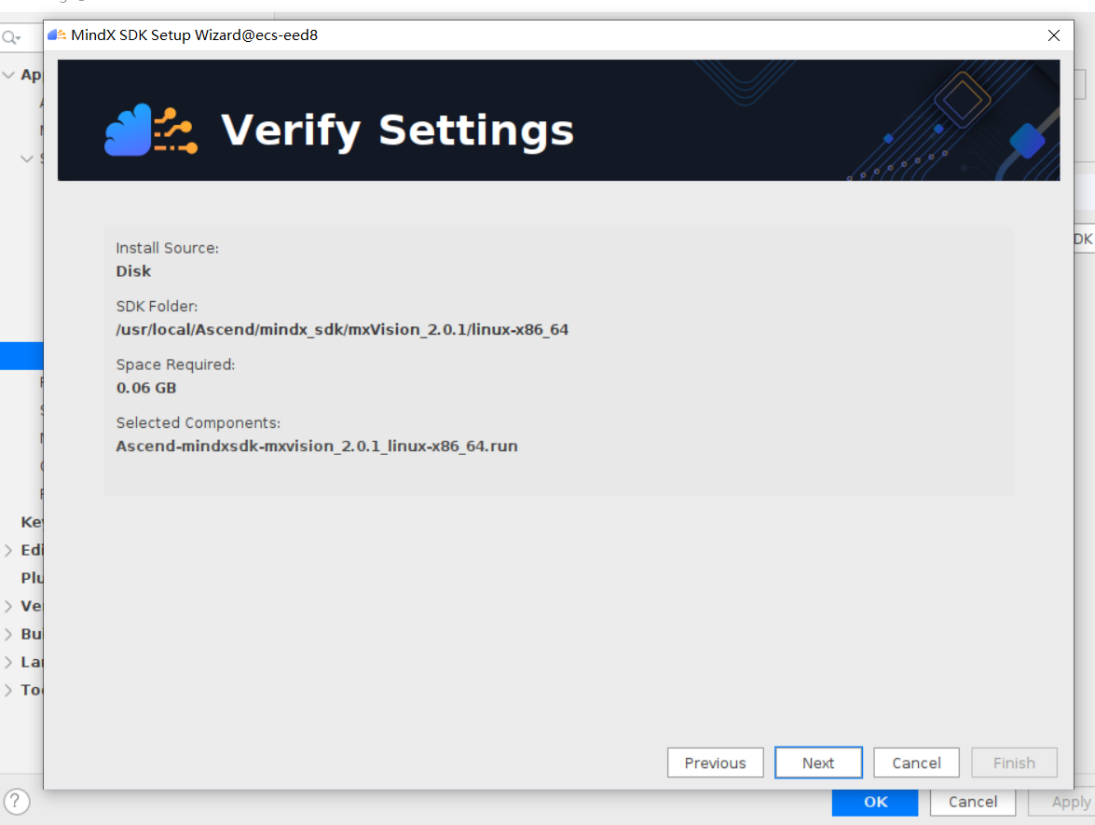
安装SDK 路径选择

安装

点击“Next”、“Next”开始安装，待提示“Installation succeeded. Please click “Finish””息，表示MindX SDK通过MindStudio安装完毕，可以在工程内使用。



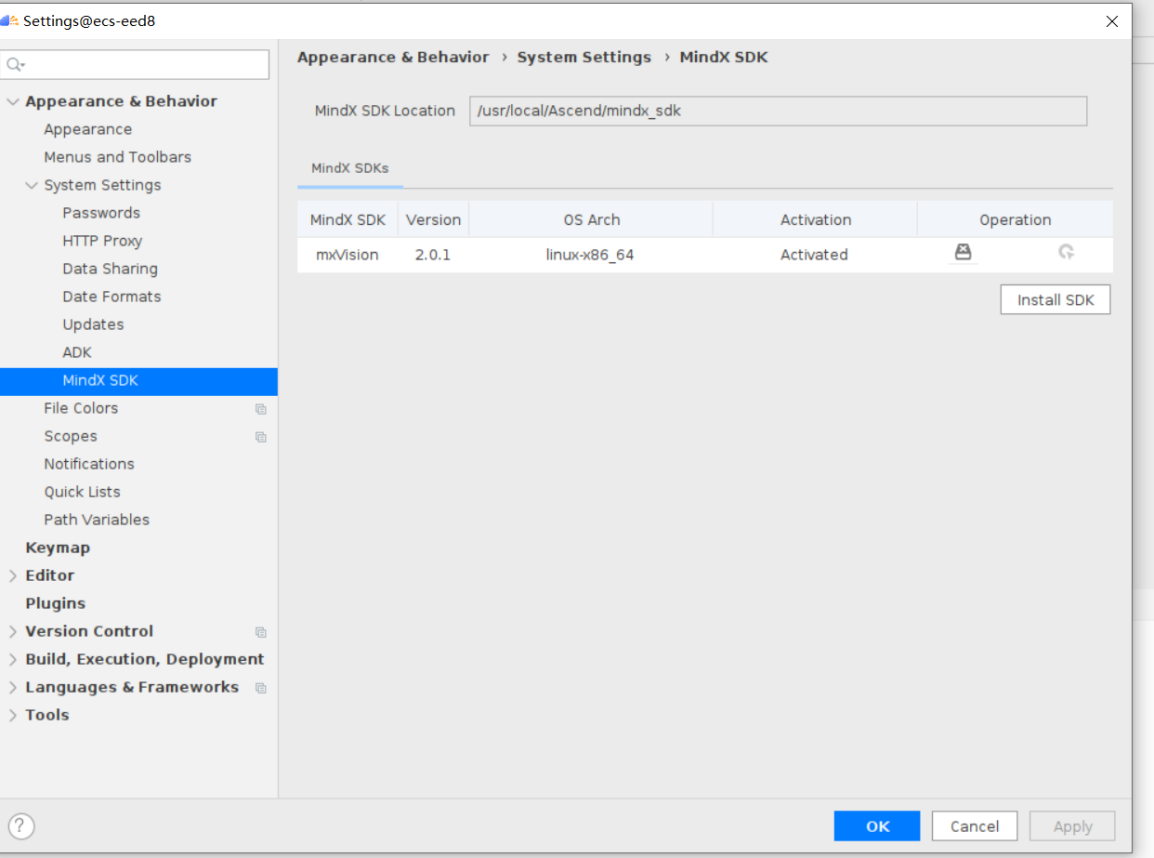
SDK 安装界面



SDK 安装确认界面



安装完成



SDK 安装完成确认面

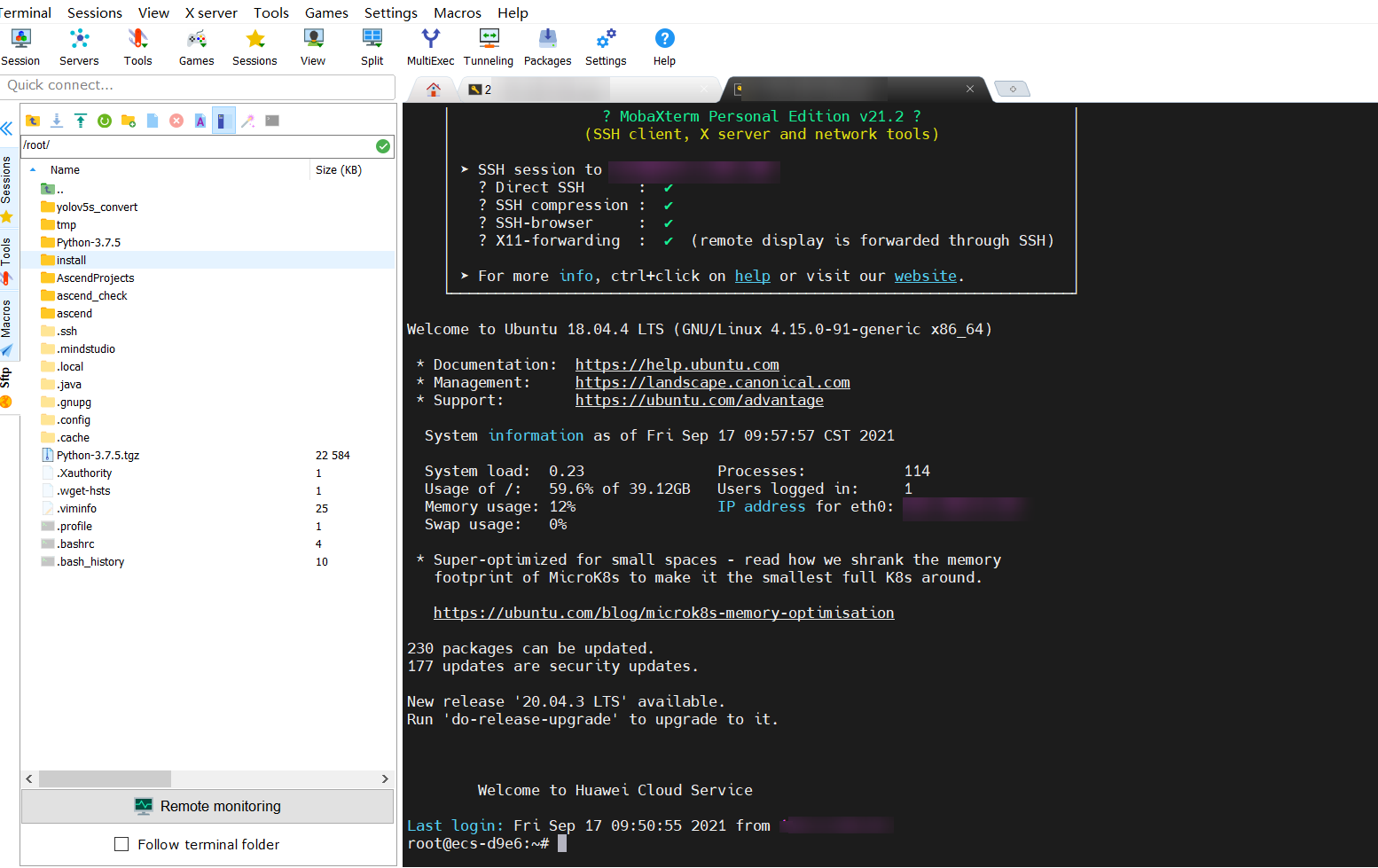
## 预置离线模型和测试图片

### 预制离线模型

在 /home/HwHiAiUser/MindX/yolov3/目录下已为您准备好了使用coco数据集训练的yolov3离线模型om文件及相关配置文件，可检测80类目标，输入分辩率为416\*416；

复制窗口

选中MobaXterm 窗口，右键选择“duplicate tab”，复制当前窗口。



打开新窗口

拷贝预置离线模型

输入如下命令，复制MindX SDK工程预置离线模型

cp -r /home/HwHiAiUser/MindX/yolov3/ /root/AscendProjects/MyApp/models/

### 预置测试图片

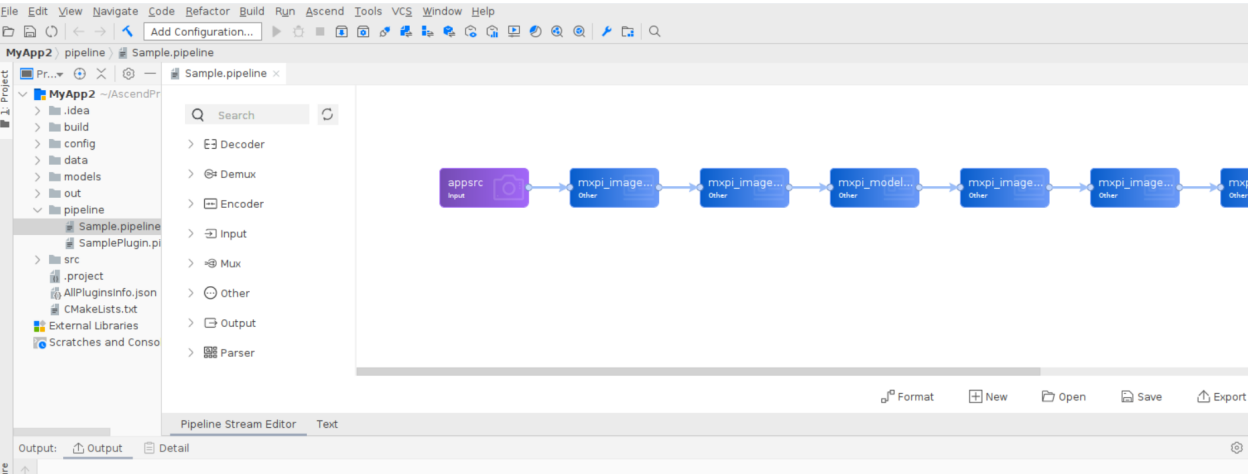
在 /root/AscendProjects/MyApp/data/目录下准备好了测试图片。



test.jpg

## 业务流程编排

在工程左侧目录结构视图中点击“MyApp”展开工程目录结构，在文件夹pipeline文件夹下双击“Sample.pipeline”文件，打开流程编排画布，在目录结构右侧窗口显示，如下图所示。



流程编排画布

该业务流为目标检测加分类的二级推理，以一级推理目标检测应用的业务流为例，我们在此二级推理业务流基础上进行重编排，操作如下:

删除三个插件

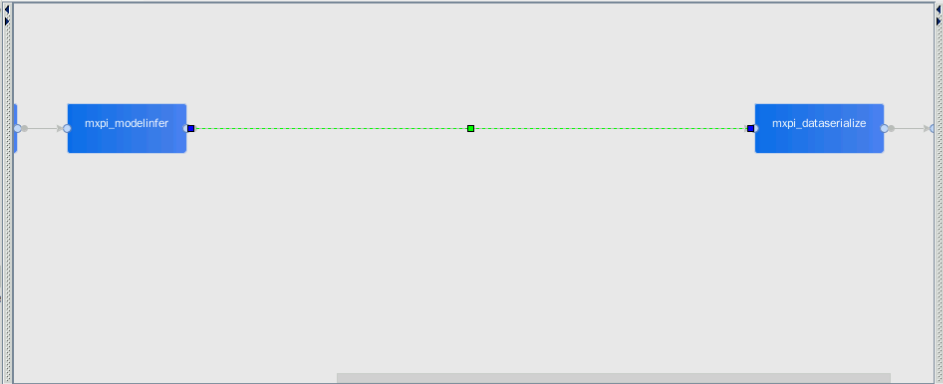
鼠标从左至右分别选择mxpi\_imagecrop0、mxpi\_imageresize1、mxpi\_modelinfer1（第二个），并按“delete”键，删去二级推理相关插件（即删除下图红框内三个插件）。



流程重编排

连接插件

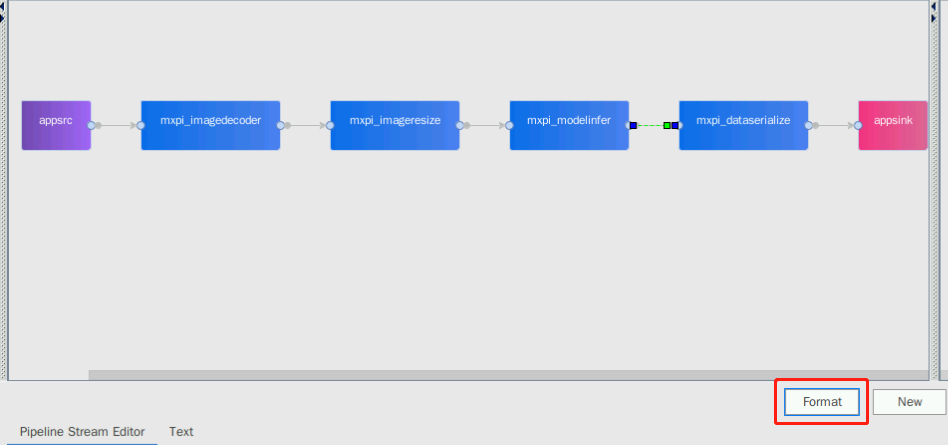
再将断点处连接起来，鼠标选择mxpi\_modelinfer0（第一个）插件右侧连接点，长按鼠标左键引出连接线，将其与mxpi\_dataserialize0插件左侧连接点相连。



流程重编排连接

格式化插件

点击画布下方“Format”键，格式化业流中将其居中显示如下图所示：



流程重编排 format

一条由数据接入插件（appsrc）、图像解码插件（mxpi\_imagedecoder0）、图像缩放插件（mxpiimageresize0）、模型推理插件（mxpi\_modelinfer0）、数据序列化插件（mxpi\_dataserialize0）和数据输出插件（appsink0）串连而成的目标检测业务流编排完毕。

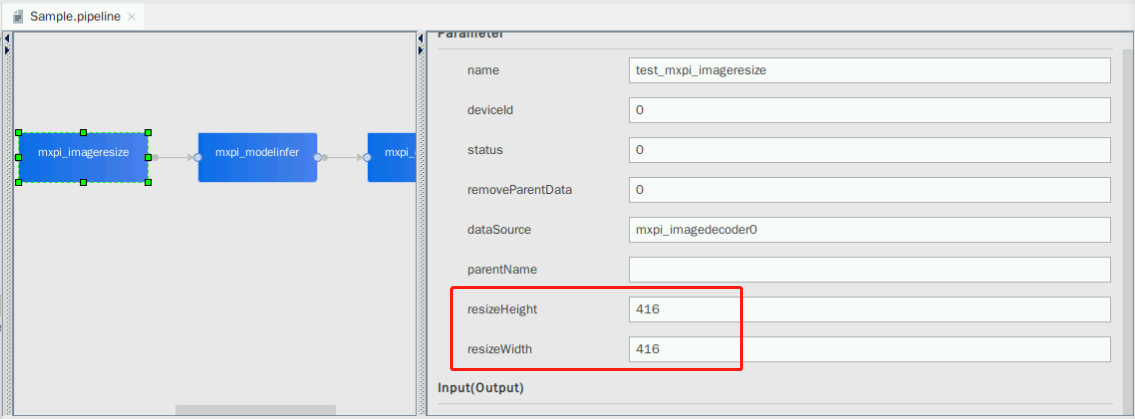
## 配置插件属性

在画布中鼠标点击插件，右侧会显示插件属性列表。接下来我们需要对一些插件的属性列表进行修改。

appsrc0：无需修改，保持原属性值。

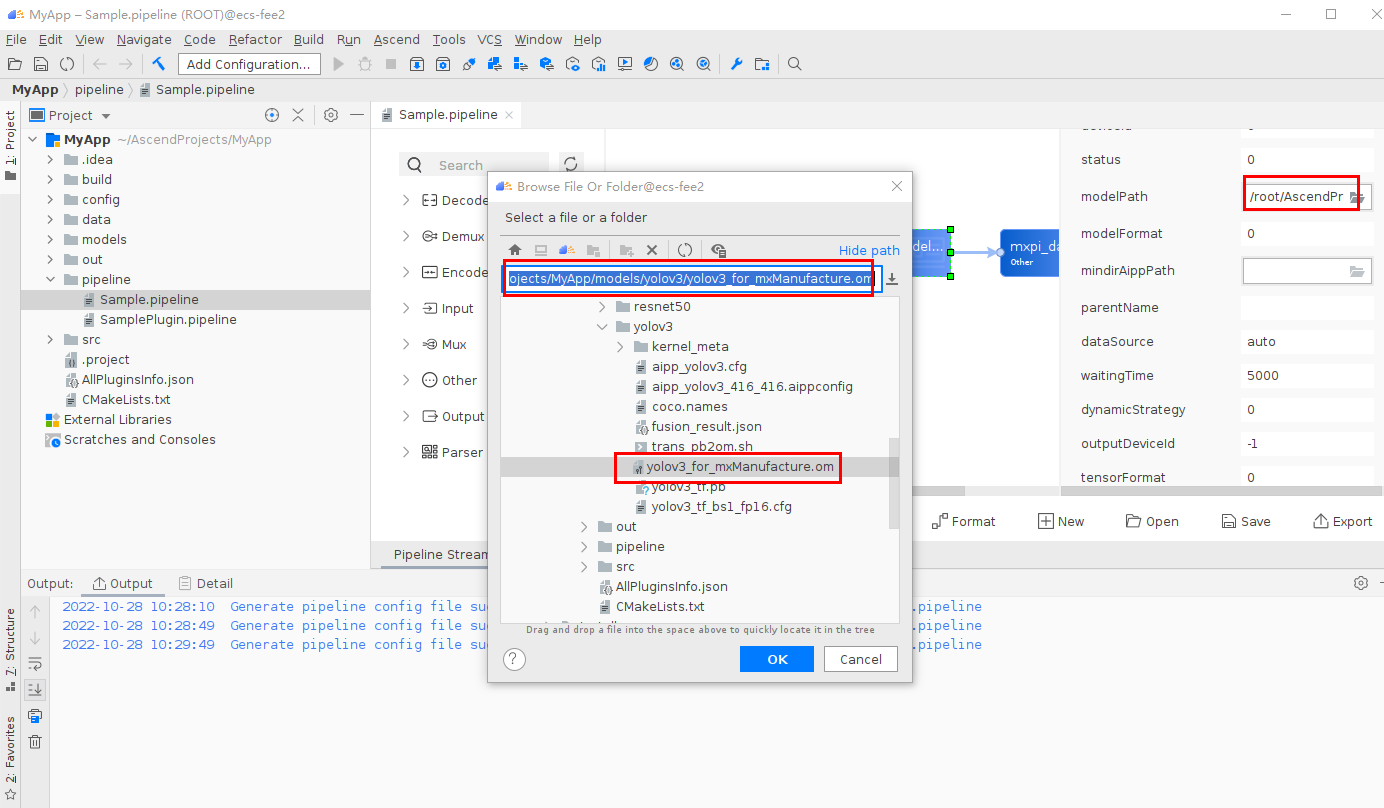
mxpi\_imagedecoder0：无需修改，保持原属性值。

mxpi\_imageresize0：在画布中鼠标点击选择mxpi\_imageresize0插件，由于我们将要使用的yolov3模型输入分辨率为416\*416，将resizeHeight和resizeWidth属性均设置为416，如下图所示。若已为416\*416则无需设置。



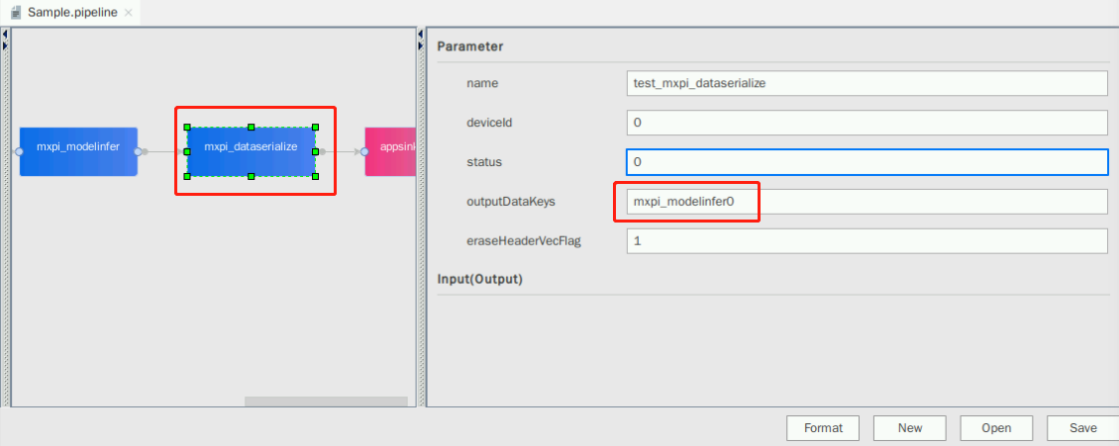
mxpi\_imageresize 设置

mxpi\_modelinfer0：在画布中鼠标点击选择mxpi\_modelinfer0插件，“modelPath”：离线模型。鼠标点击modelPath属性右侧的文件夹选项，选择模型路径为：/root/AscendProjects/MyApp/models/yolov3/yolov3\_for\_mxManufature.om



mxpi\_modelinfer 模型路径选择

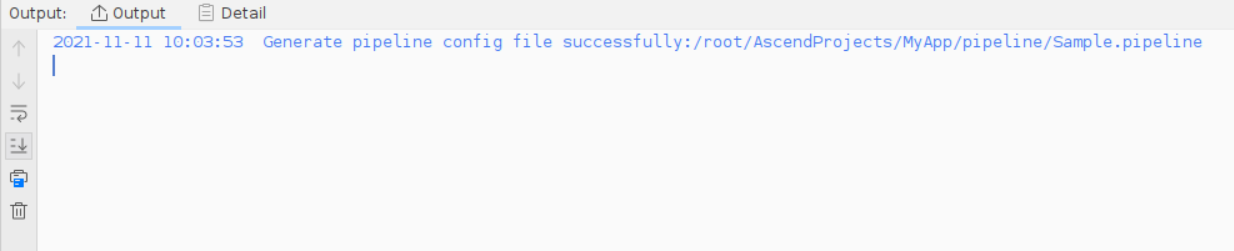
mxpi\_dataserialize0：在画布中鼠标点击选择mxpidataserialize0插件，outputDataKeys属性删除”,mxpi\_modelinfer1”，保留mxpi\_modelinfer0，仅序列化输出一级推理插件的推理结果



mxpi\_dataserialize 设置

appsink：无需修改，保持原属性值。

依次点击Save->Save->Replace，保存pipeline。



## 编写应用代码

MindStudio工程已为用户自动生成了样例应用代码，参考工程目录结构中的”MyApp->src->main.cpp“文件和同级目录的CMakeLists.txt文件。需要注意的是，main.cpp文件中指定了应用所使用的pipeline文件为我们编排好的Sample.pipeline文件，所用的图片为”MyApp->data->test.jpg“，如下图所示。实际开发时，用户可自行指定pipeline文件和测试图片。

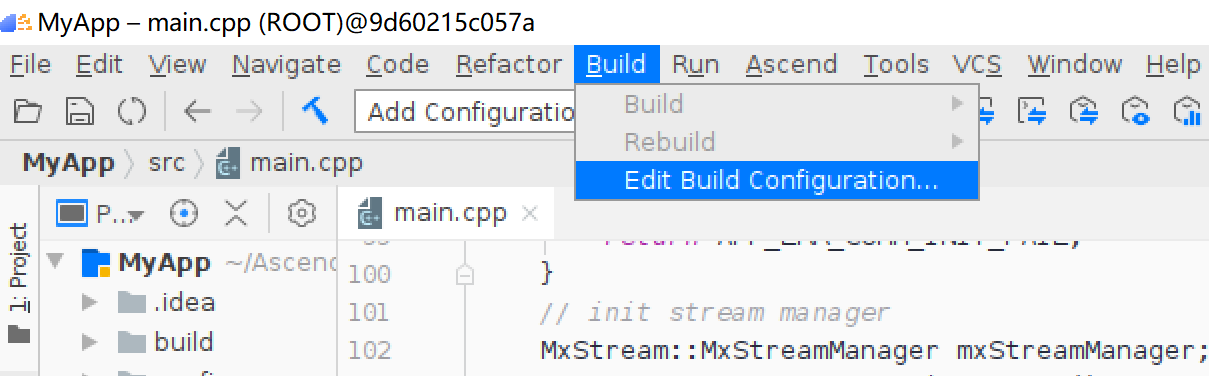


Main.cpp

## 编译运行

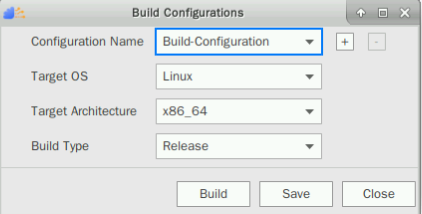
### 编译

在MindStudio顶部菜单栏点击”Build“->"Edit Build Configuration..."，打开设置页面如下图所示。



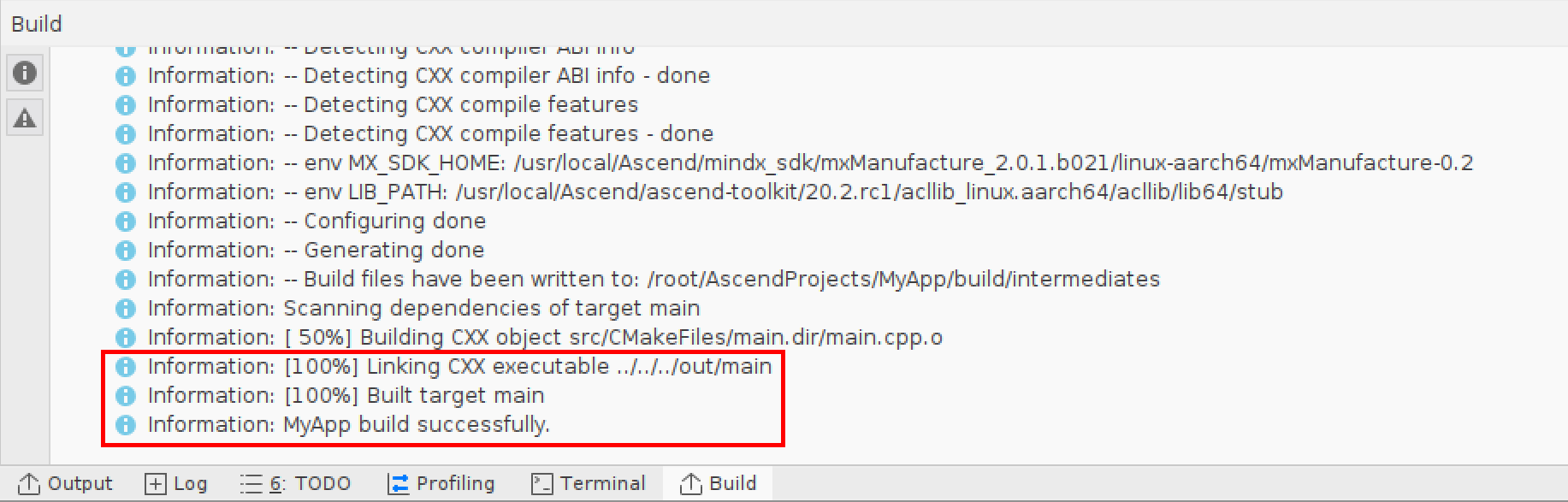
编译界面

确认参数如下图所示。



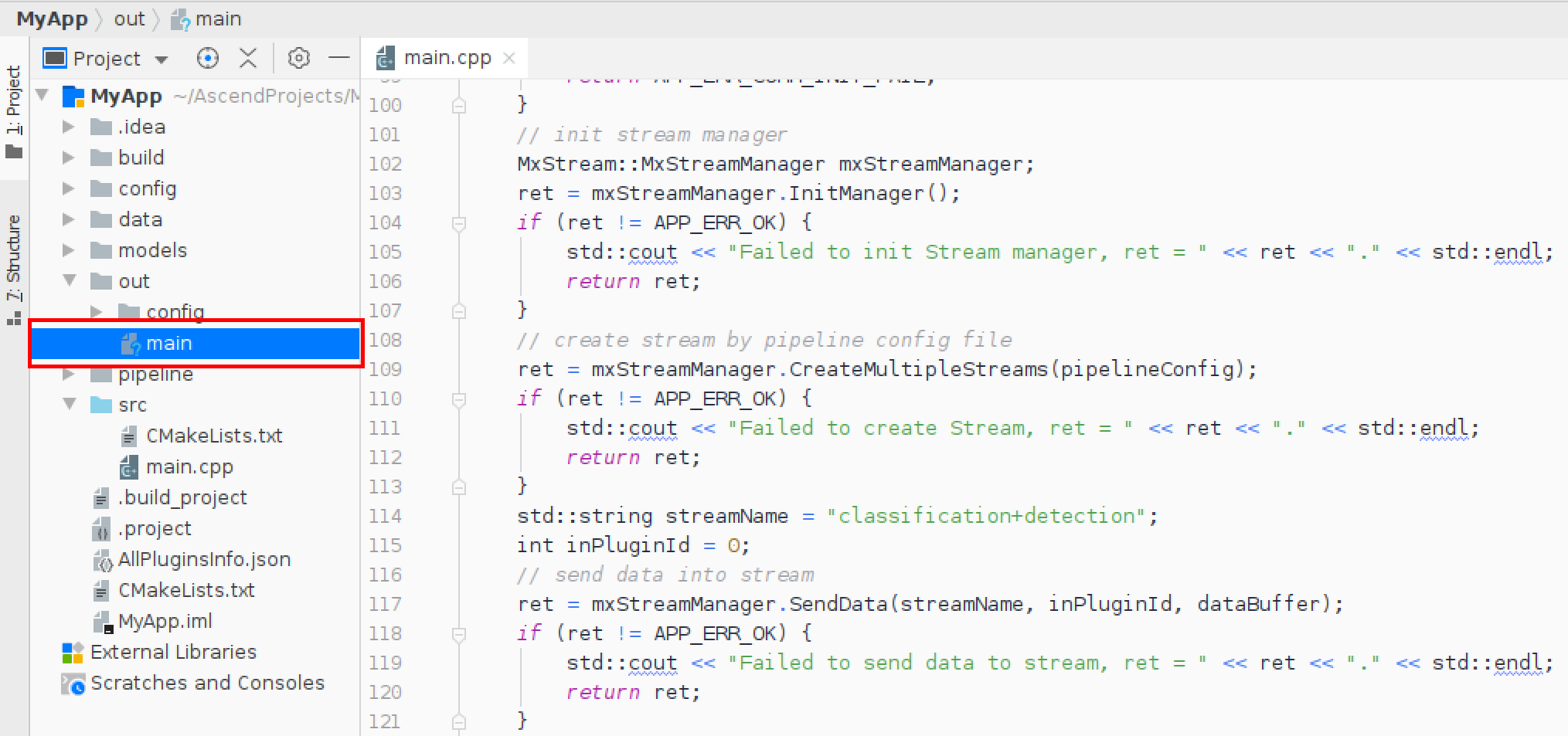
编译参数

点击”Build“编译成功，如下图所示。



编译成功界面

此时，在"MyApp/out"目录下生成可执行文件main，如下图所示



可执行文件main 路径

### 运行

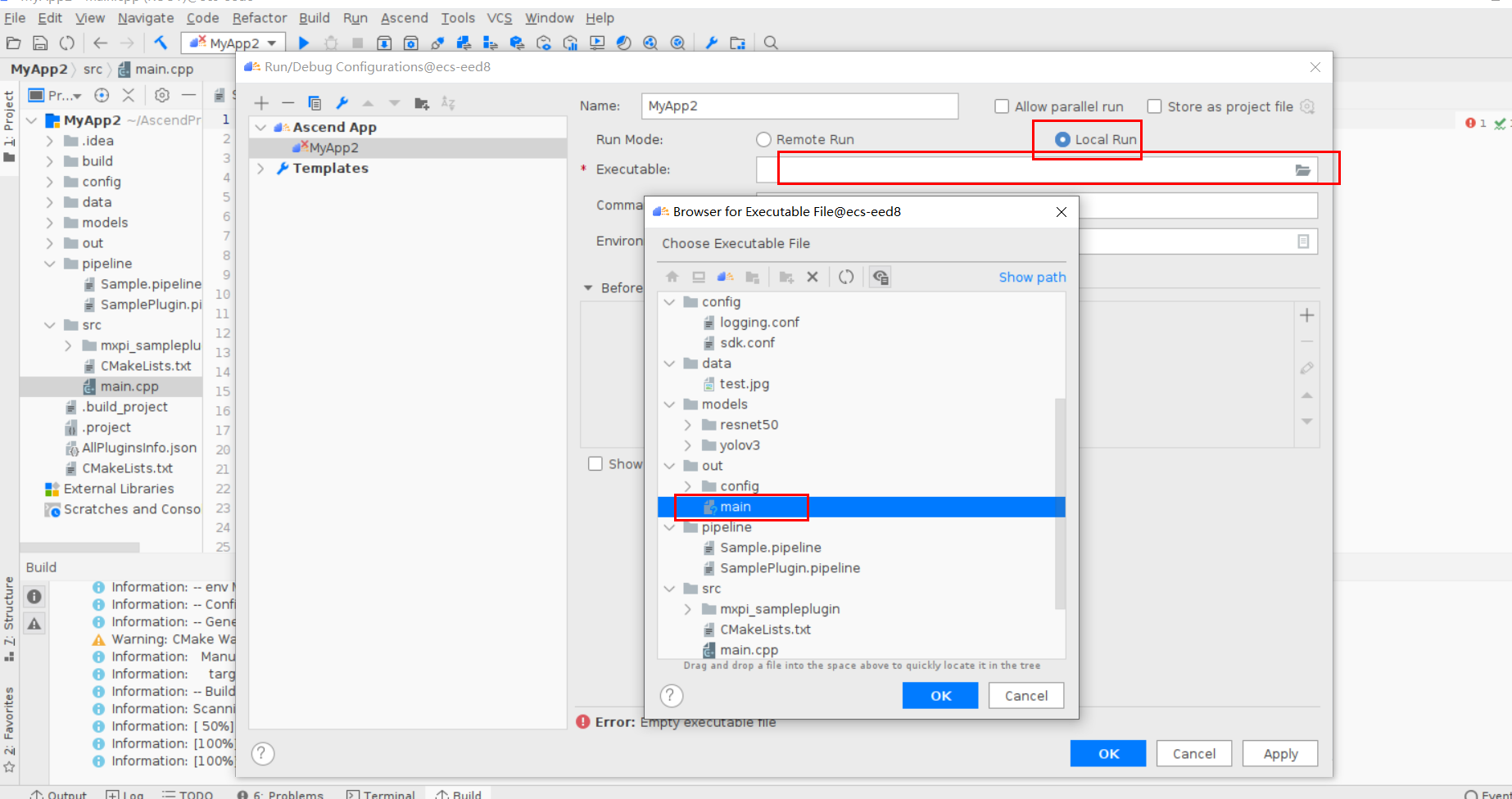
在MindStudio顶部菜单栏点击"Run"->"Run..."->"Edit Configuration..."，在弹出的运行配置窗口设置如下：

Name MyApp（可修改）

Run Mode Local Run

Excutable /root /AscendProjects/MyApp/out/main（绝对路径）

如下图所示：



可执行文件

点击Run按钮

在MindStudio底部输出如下运行日志，表示目标检测成功，图片中目录为dog，检测结果置信度为0.99774956699999995，检测框坐标为："x0":148.73291,"x1":886.13012700000002,"y0":138.64837600000001,"y1":607.08746299999996，如下图所示：



打印日志

## 结果可视化

### 修改代码

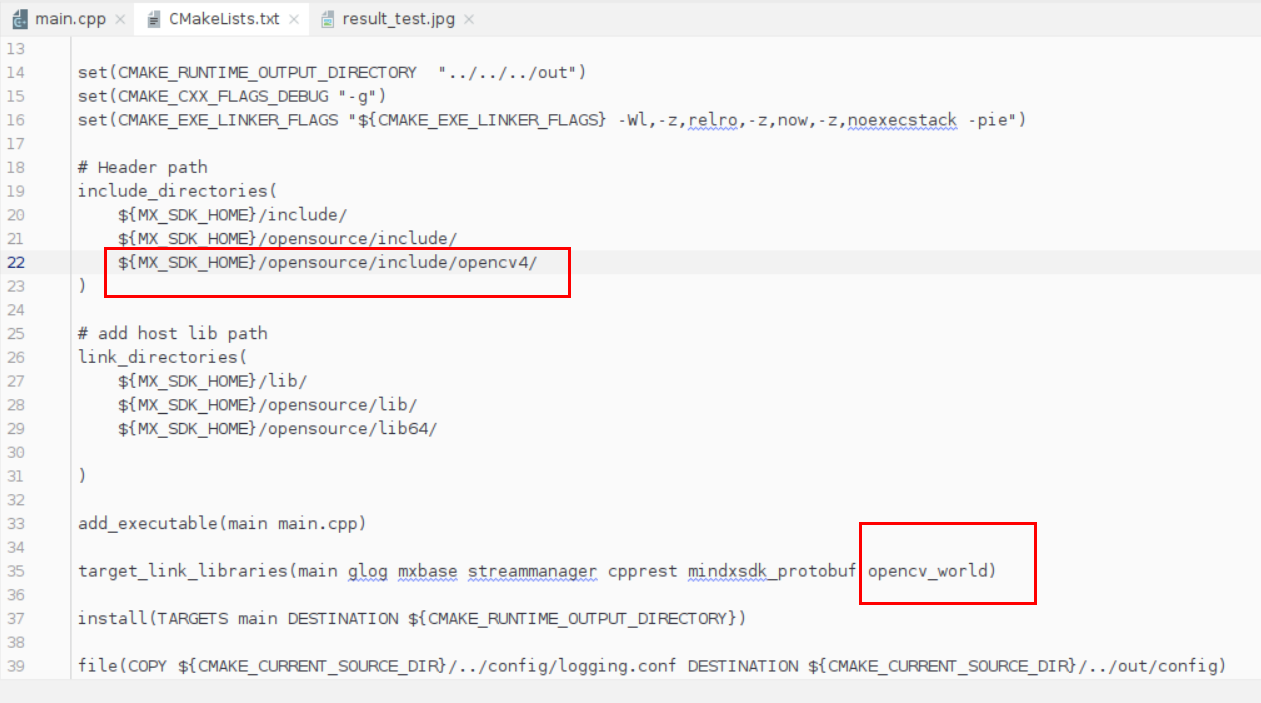
修改CMakeLists.txt文件

在MindStudio左侧目录结构视图中点击“MyApp->src->CMakeList.txt”文件，在第22行插入以下代码 ：

${MX\_SDK\_HOME}/opensource/include/opencv4/

在第34行加上以下代码：

opencv\_world



修改后

修改main.cpp文件

在MindStudio左侧目录结构视图中点击“MyApp->src->main.cpp”文件，在第19行后插入以下代码：

#include "opencv4/opencv2/opencv.hpp"

在第131行插入以下代码块：

web::json::value jsonText = web::json::value::parse(result);

if (jsonText.is\_object()) {

web::json::object textObject = jsonText.as\_object();

auto itInferObject = textObject.find("MxpiObject");

if (itInferObject == textObject.end() || (!itInferObject->second.is\_array())) {

return 0;

}

auto iter = itInferObject->second.as\_array().begin();

cv::Mat src = cv::imread("../data/test.jpg");

for (; iter != itInferObject->second.as\_array().end(); iter++) {

if (iter->is\_object()) {

auto modelInferObject = iter->as\_object();

float x0 = 0;

float x1 = 0;

float y0 = 0;

float y1 = 0;

auto it = modelInferObject.find("x0");

if (it != modelInferObject.end()) {

x0 = float(it->second.as\_double());

}

it = modelInferObject.find("x1");

if (it != modelInferObject.end()) {

x1 = float(it->second.as\_double());

}

it = modelInferObject.find("y0");

if (it != modelInferObject.end()) {

y0 = float(it->second.as\_double());

}

it = modelInferObject.find("y1");

if (it != modelInferObject.end()) {

y1 = float(it->second.as\_double());

}

cv::Rect rect(x0, y0, x1 - x0, y1 - y0);

cv::rectangle(src, rect, cv::Scalar(0, 255, 0),5, cv::LINE\_8,0);

}

}

cv::imwrite("./result\_test.jpg", src);

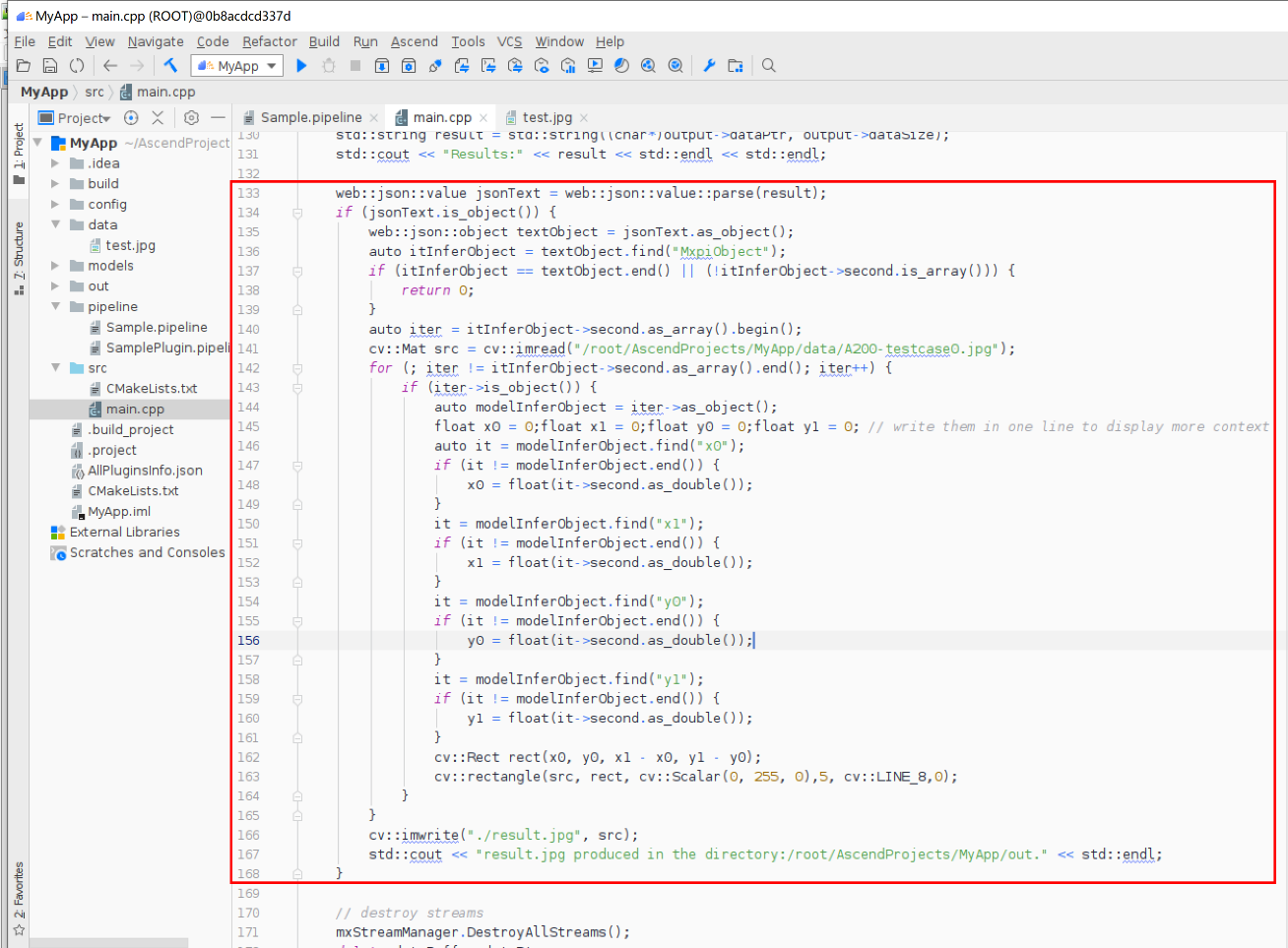
std::cout << "result\_test.jpg produced in the directory:/home/user/AscendProjects/MyApp/out." << std::endl;

}

添加成功，保存，最终如下两图所示：



Main.cpp 修改

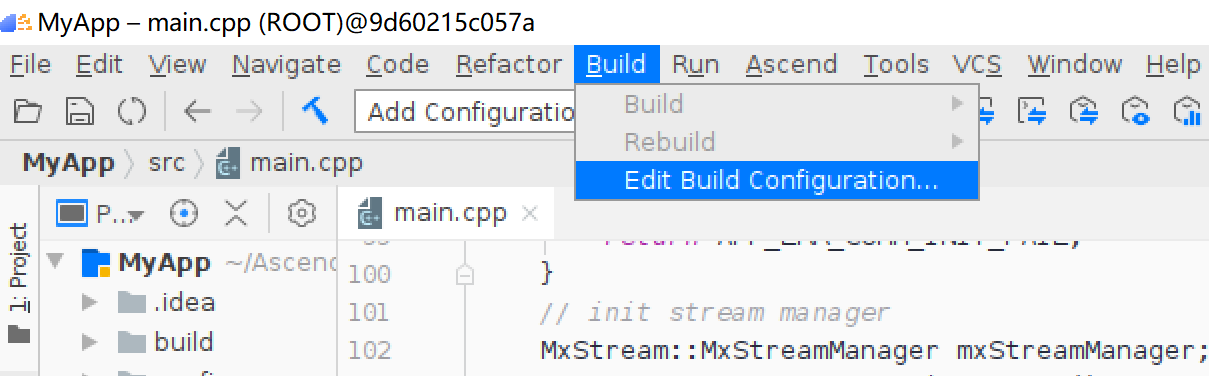


Main.cpp 修改

**注意：根据代码上下文确认插入位置是否正确。**

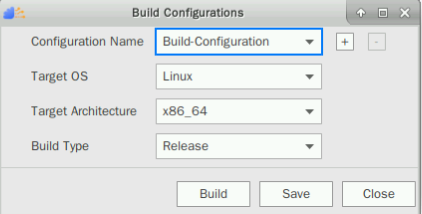
### 编译

在MindStudio顶部菜单栏点击”Build“->"Edit Build Configuration..."，打开设置页面如下图所示：



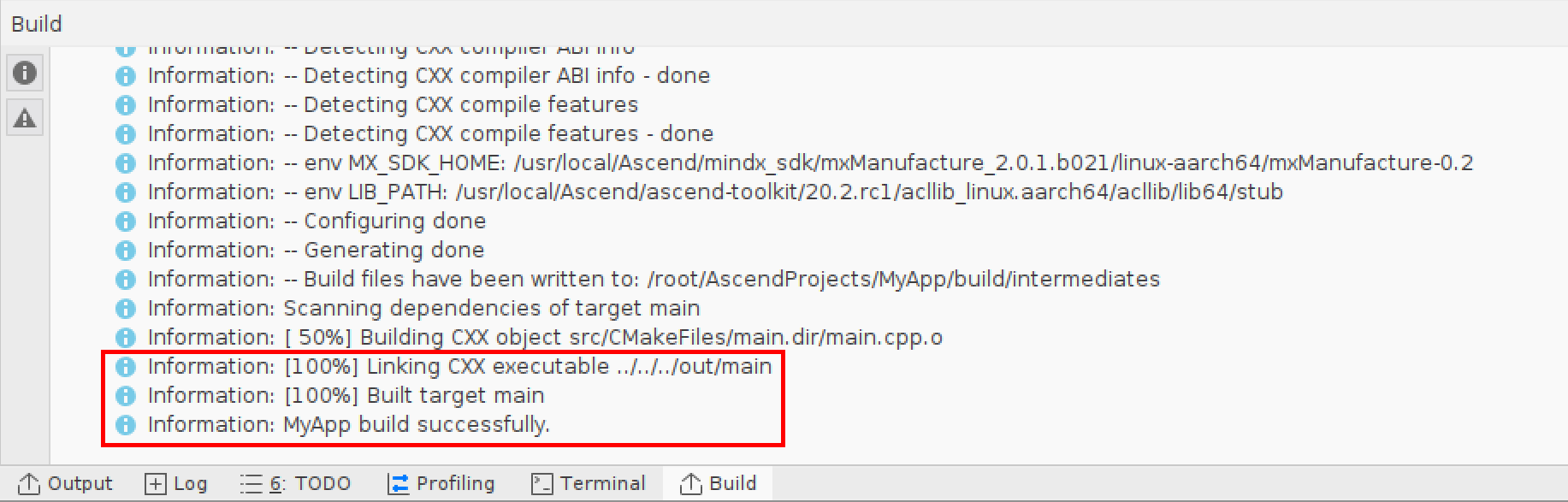
编译界面

确认参数如下图所示。



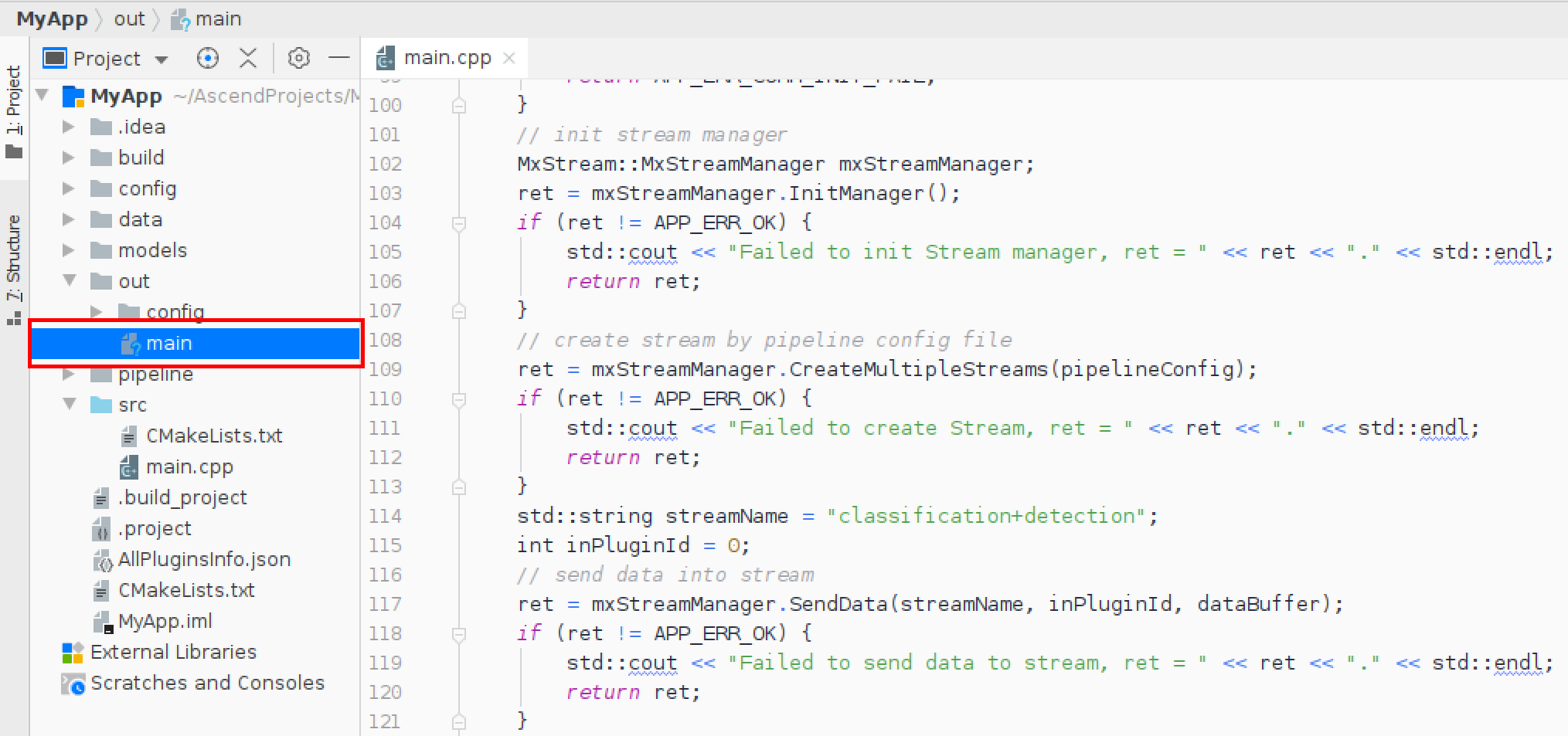
编译参数

点击”Build“编译成功，如下图所示：



编译成功界面

此时，在"MyApp/out"目录下生成可执行文件main，如下图所示：



可执行文件main 路径

### 运行

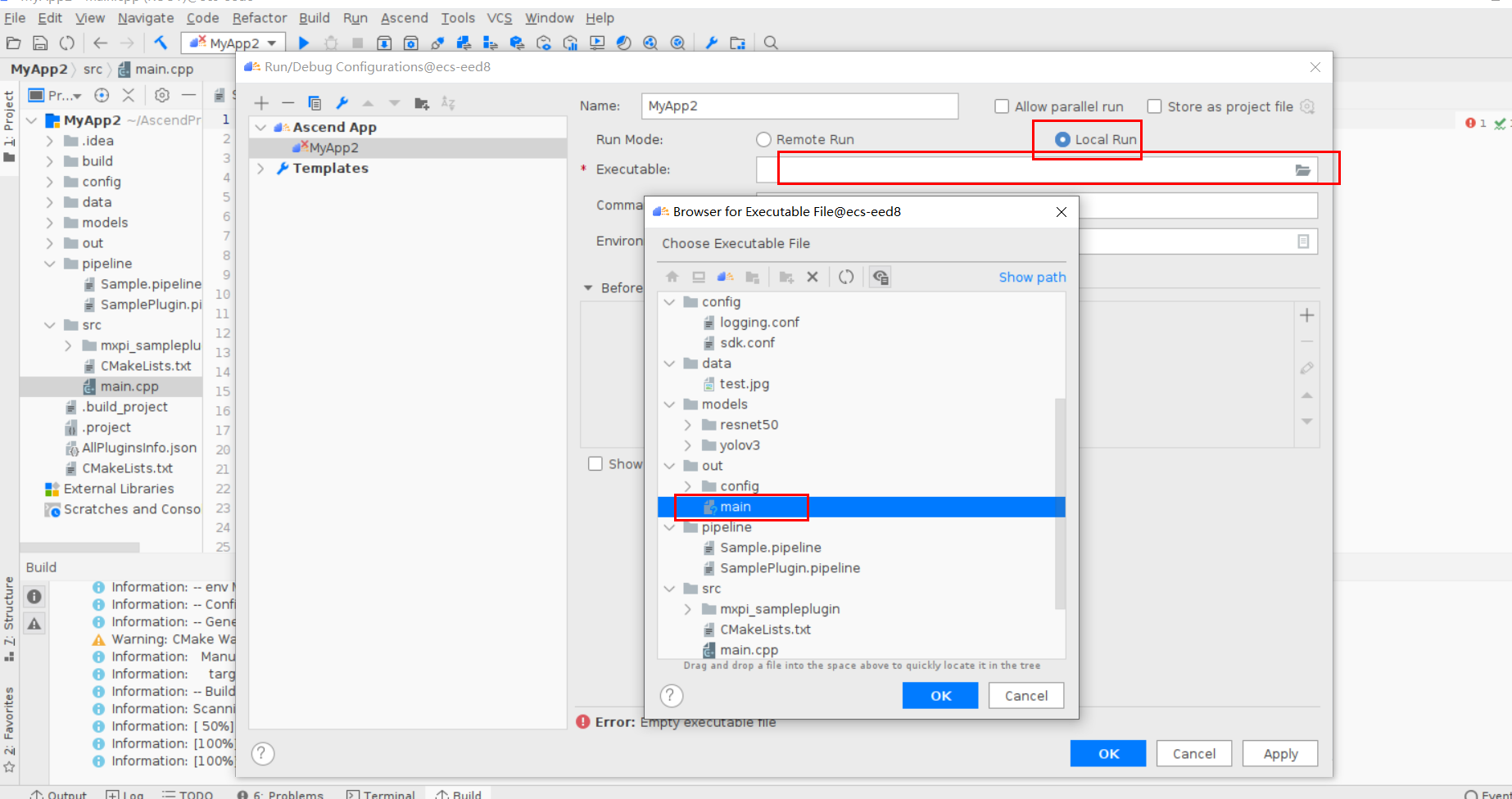
在MindStudio顶部菜单栏点击"Run"->"Run..."->"Edit Configuration..."，在弹出的运行配置窗口设置如下：

Name MyApp（可修改）

Run Mode Local Run

Excutable /root /AscendProjects/MyApp/out/main（绝对路径）

如下图所示：



可执行文件

点击Run按钮。

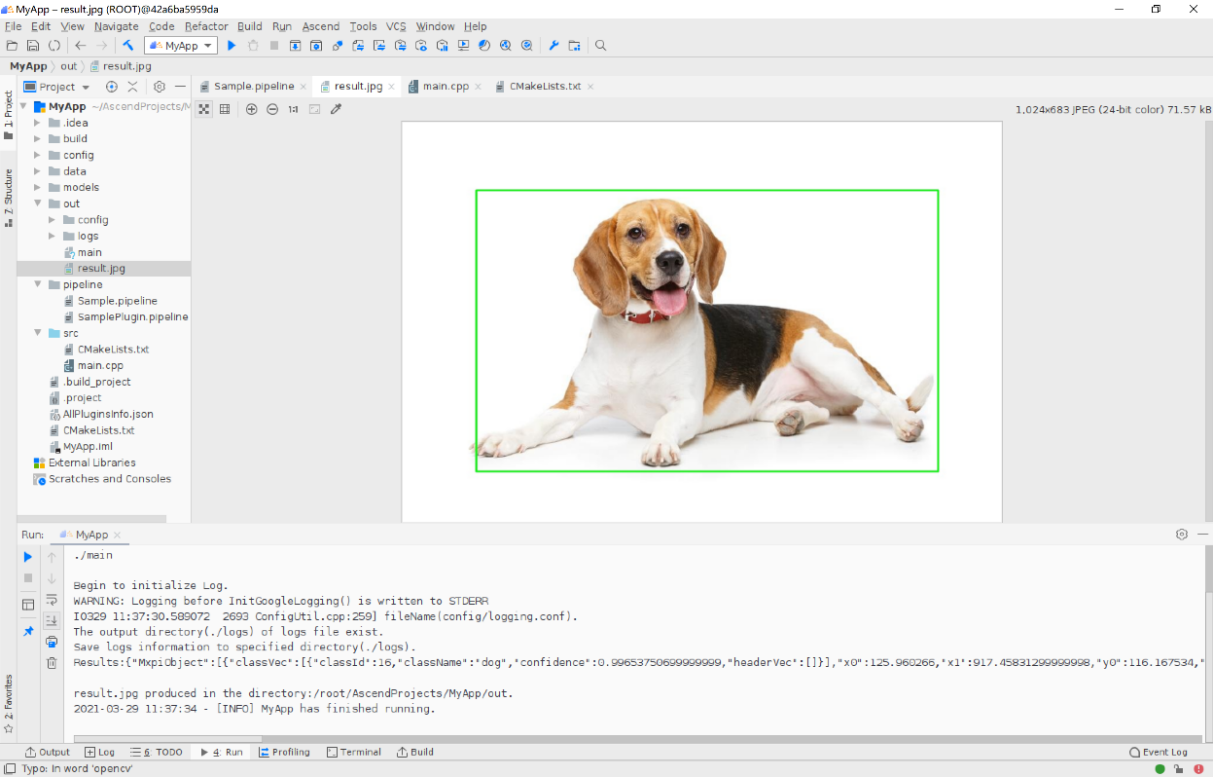
点击Run按钮

在MindStudio底部输出如下运行日志，表示目标检测成功，图片中目录为dog，检测结果置信度为0.99774956699999995，检测框坐标为："x0":148.73291,"x1":886.13012700000002,"y0":138.64837600000001,"y1":607.08746299999996，如下图所示：



打印日志

可视化结果存放在MyApp->out目录下，为result\_test.jpg文件，如下。



result\_test.jpg

## 实验小结

本实验通过环境搭建、业务流编排、配置插件属性、编写应用代码、离线模型和模型和图片、结果可视化全流程展示如何使用MindX SDK开发目标检测应用。

# 附录：环境准备

## 获取镜像

本节指导如何获取华为云账号下项目ID，以便方便导入实验所需镜像。

获取项目ID

进入华为云网址：<https://www.huaweicloud.com/>

点击 “账号中心”



账号中心

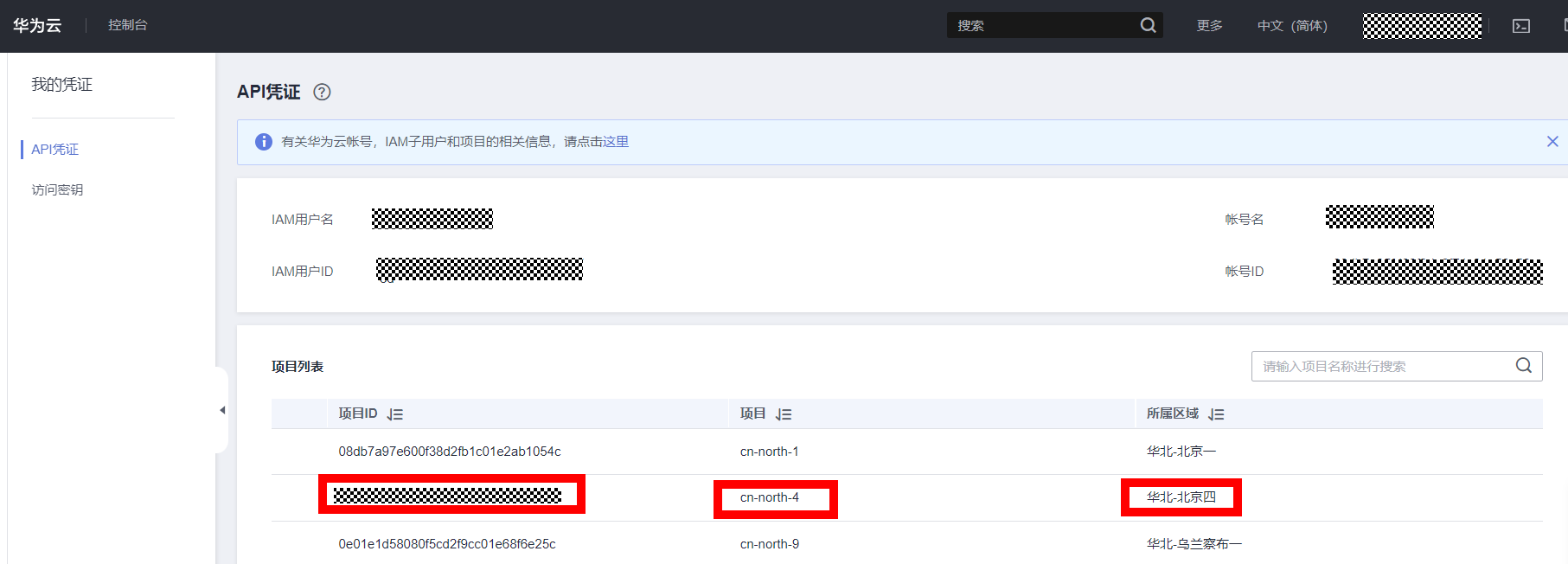
点击“我的凭证”



我的凭证

获取项目ID

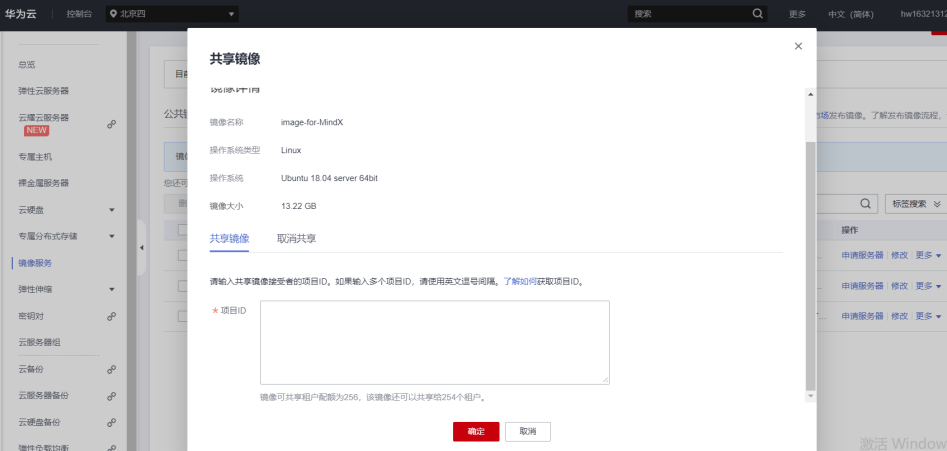
选择项目为“cn-north-4”的项目ID，[高校教师通过邮件发送至yanxukun1@huawei.com](mailto:高校教师通过邮件发送至yanxukun1@huawei.com ) ,华为技术支持给高校老师分发镜像 “image-for-MindX”；若学生则需要自行发送给任课老师，由老师发放。



项目ID

共享镜像

（镜像分享者需要操作）把项目ID复制到项目ID 里,点击确定。



共享镜像

## 创建ECS 弹性服务器

### 创建ECS弹性云服务器

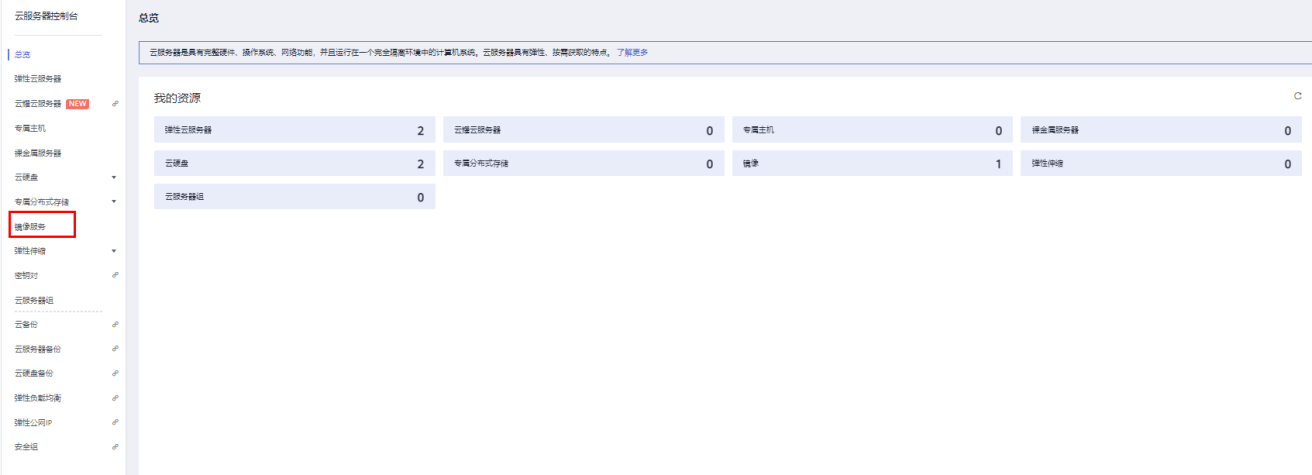
进入华为云ECS控制台

在[华为云ECS主页](https://www.huaweicloud.com/product/ecs.html)，点击“管理控制台”进入ECS的管理页面。



华为云ECS主页

接受共享镜像

在左侧菜单栏中选择“镜像服务”

镜像服务

选择“共享镜像”，“全部接受”。

共享镜像

创建弹性云服务器

控制台区域选择“华北－北京四”，在左侧菜单栏中选择“弹性云服务器”，在右上角“购买弹性云服务器”。



ECS 控制台

在“基础配置”里，选择如下配置：

计费模式：按需计费。

区域：华北-北京四。

可用区：随机分配。

CPU架构：x86计算。

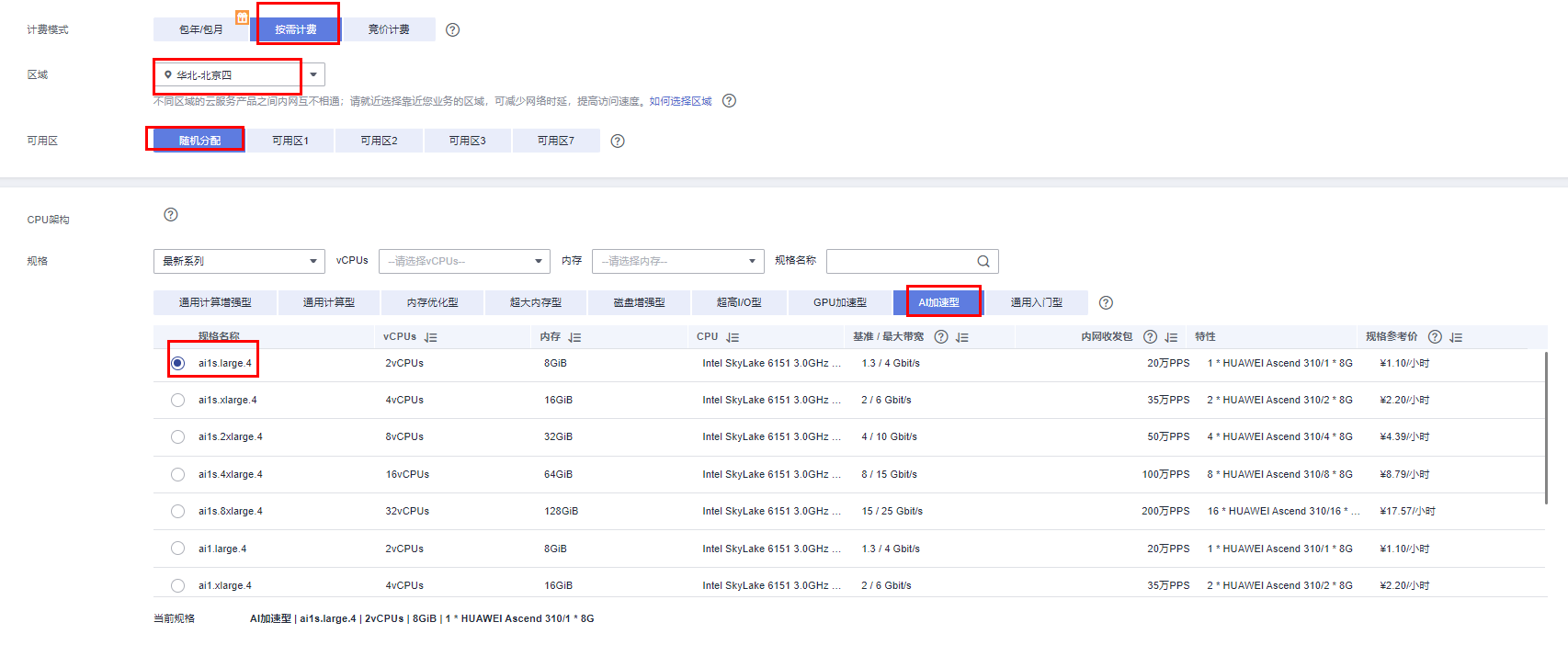
规格： AI加速型 | ai1s.large.4 | 2vCPUs | 8GiB | 1 \* HUAWEI Ascend 310/1 \* 8G。

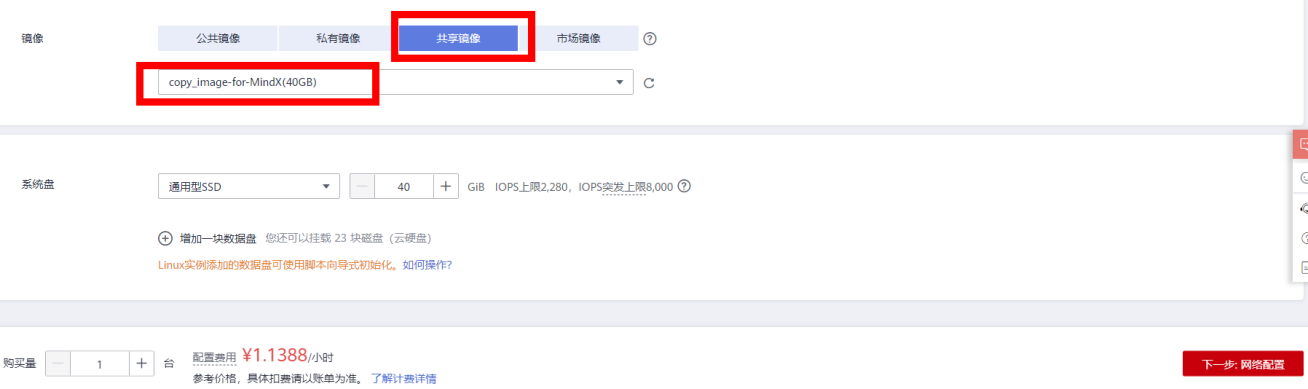
镜像：公享镜像，image-for-MindX。

规格：Ascend: 1\*Ascend910|CPU: 24核 96GB。

系统盘：通用型SSD，40GB。

如图所示：





ECS基础配置

基础配置完成之后，选择“下一步:网络配置”。

在“网络配置”里，选择如下配置：

网络：可以前往控制台创建新的虚拟私有云。

拓展网卡：无。

安全组：可以新建安全组。

弹性公网IP：现在购买。

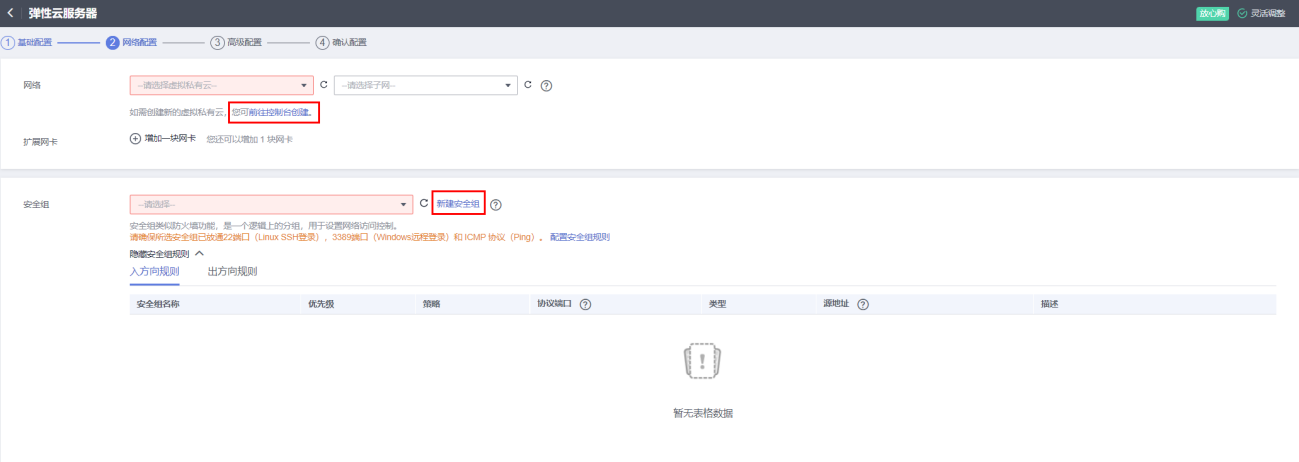
线路：全动态BGP。

公网带宽：按流量计费。

宽带大小：自定义，200Mbit/s。

释放行为：勾选随实例释放。

如图所示：





ECS网络配置

网络配置完成之后，选择“下一步”，进行高级配置。

在“高级配置”里，选择如下配置：

云服务器名称：可以自定义。

登录凭证：密码。

用户名：root。

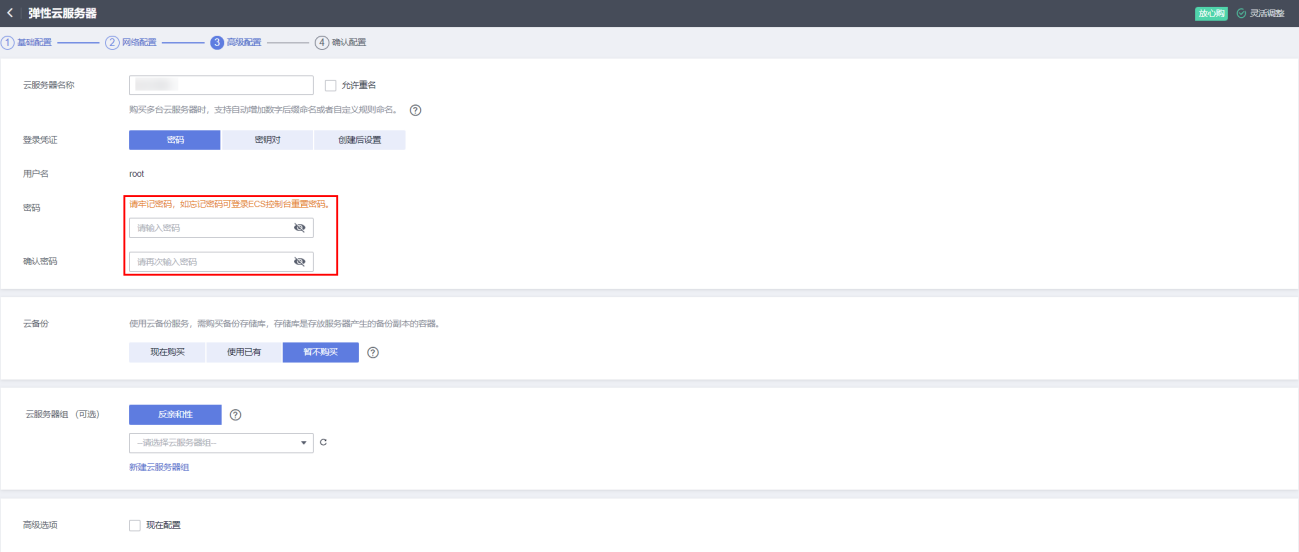
密码：自定义（后续登录使用，需谨记）。

云备份：暂不购买。

云服务器组：无。

高级选项：无。

如图所示：

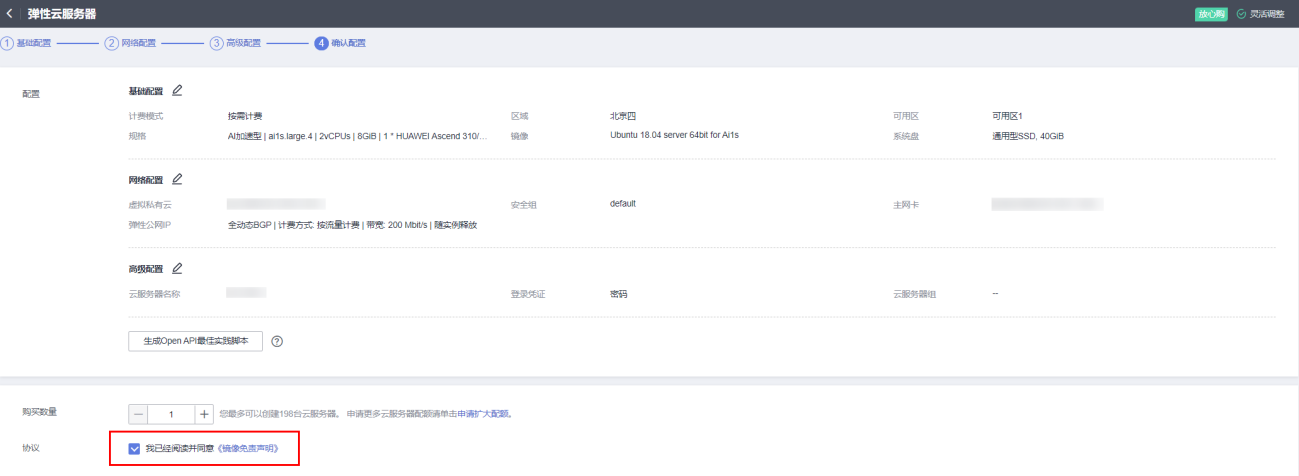


ECS高级配置

高级配置完成之后，选择“下一步”，确认配置。

在“确认配置”里，选择如下配置：

协议：勾选我已阅读并同意《镜像免责声明》。



ECS 确认配置

确认配置完成之后，选择“立即购买”。

“任务提交成功”之后，选择“返回服务器列表”即可回到弹性云服务器的管理控制台，看到已创建的ECS弹性云服务器正在运行中。

\* 注意在“IP地址”显示的弹性公网IP地址，后续会用到。



ECS弹性云服务器创建成功

### MobaXterm连接ECS

下载MobaXterm （21.2及以上版本皆可）

进入MobaXterm 的官网主页：**<https://mobaxterm.mobatek.net/>**

选择“Home Edition”，下载“MobaXterm Home Edition v21.x （Portable edition）”。

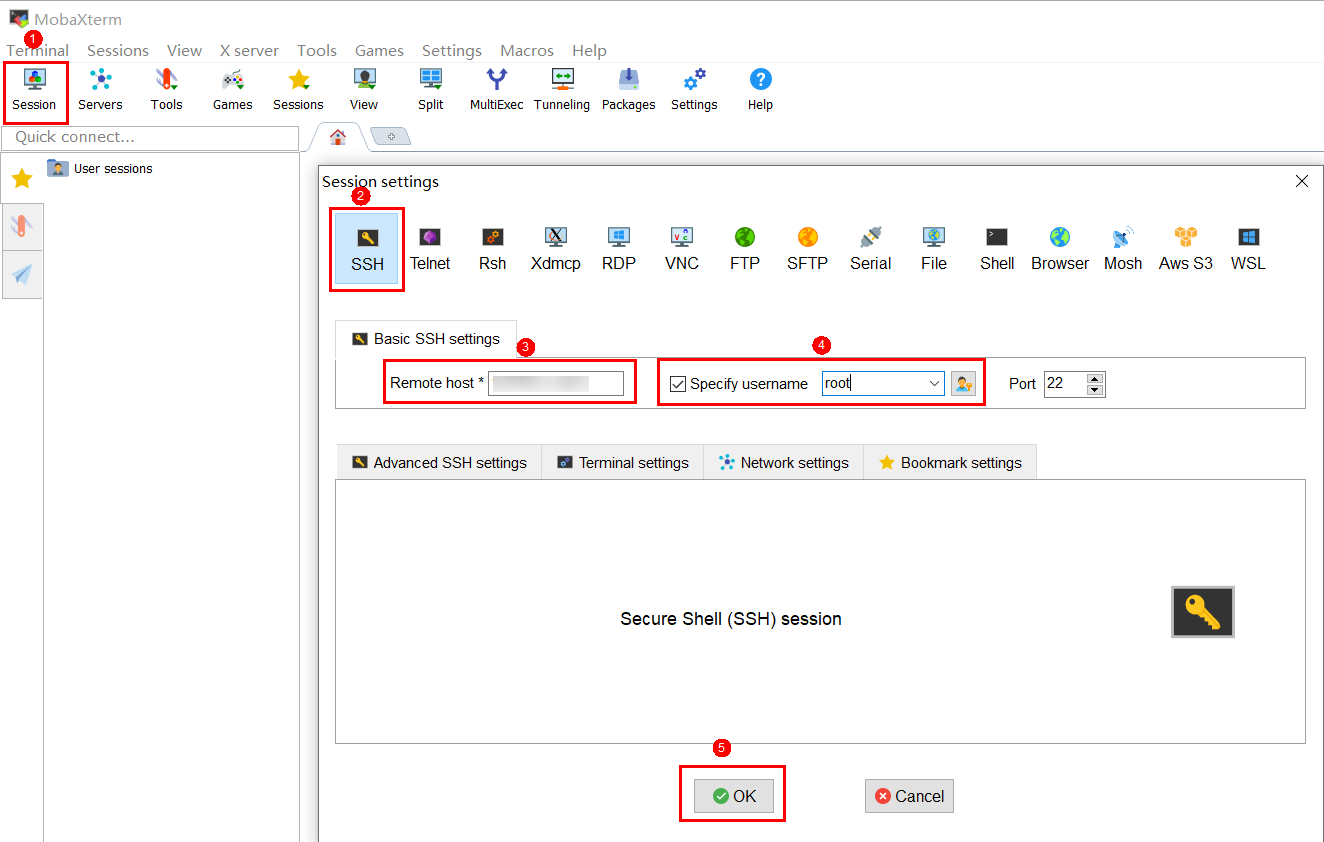
下载完成之后解压MobaXterm\_Portable\_v21.x.zip文件。



下载界面

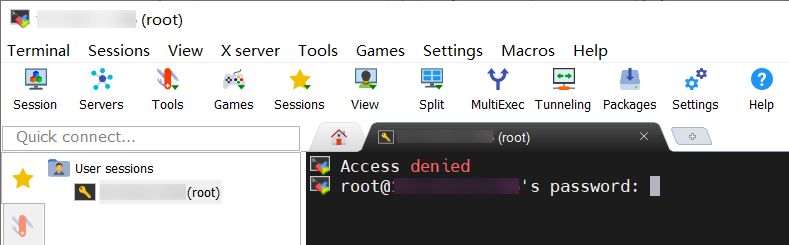
使用MobaXterm远程连接弹性云服务器

进入解压后的MobaXterm\_Portable\_v21.x文件夹，打开MobaXterm\_Personal\_21.x.exe文件，选择菜单栏的“Session”，之后进入“Session settings”页面，远程链接选择“SSH”协议，输入图3-12 ECS弹性云服务器创建成功时显示的弹性公网IP地址，选择指定用户名“Specify username”，用户名为“root”，配置完成之后选择“OK”提交。



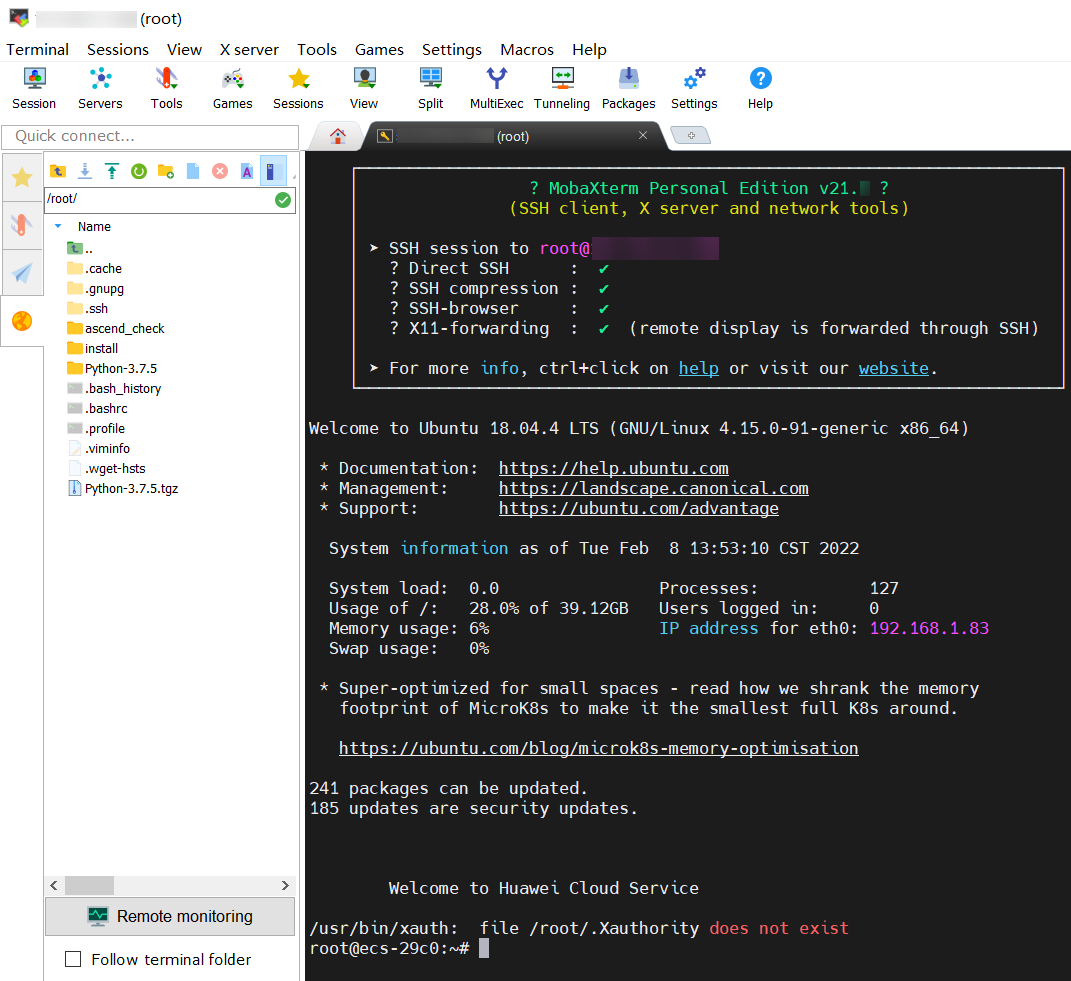
MobaXterm远程连接

MobaXterm 登录ECS需输入密码，[在3.2.1 创建ECS弹性云服务器](#_创建ECS弹性云服务器)的步骤3中，高级配置里已自定义了弹性云服务器root用户密码，在此输入即可。



MobaXterm 登录ECS需输入密码

MobaXterm远程链接弹性云服务器成功，后续还需进一步配置弹性云服务器的云上环境。



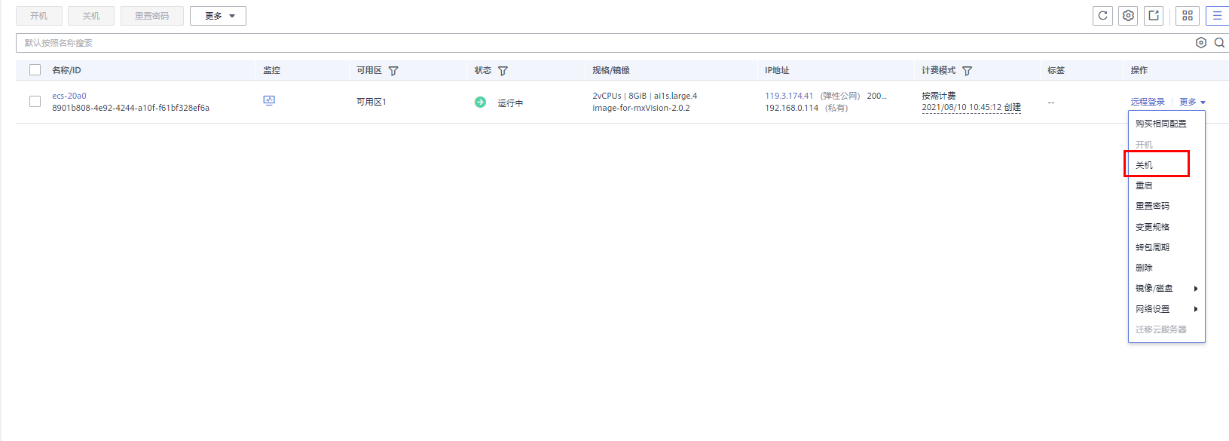
MobaXterm 登录成功界面

## 关闭ECS弹性云服务器

实验完成后，请及时关闭服务器，防止资源浪费。

关闭服务器

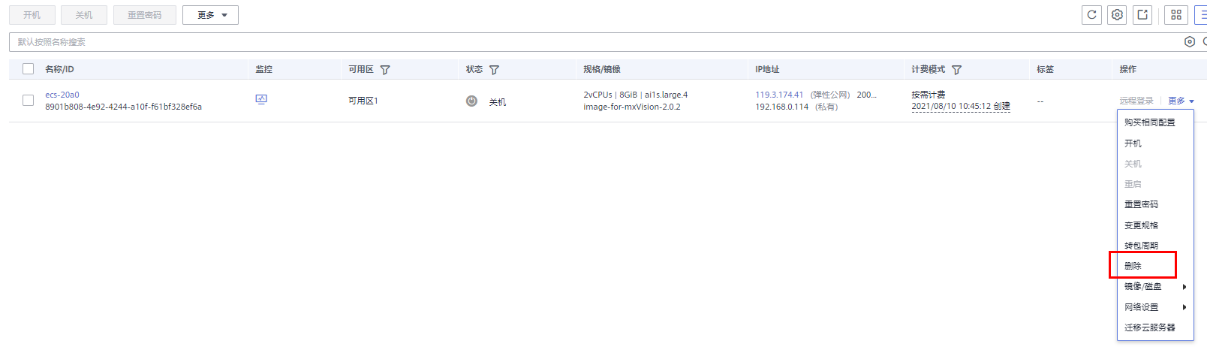
点击“更多”，选择“关机”。



关机界面

删除服务器

完成实验后，选择删除服务器，释放资源。



删除服务器



删除资源